For office use only	Team Control Number	For office use only
T1 __________	**57566**	F1 __________
T2 __________		F2 __________
T3 __________	Problem Chosen	F3 __________
T4 __________	**B**	F4 __________

2017
MCM/ICM
Summary Sheet

An optimized design of the area following the toll barrier

Summary

In order to determine the optimal area following the toll barrier, this paper proposes a function model for the turnpike authority in favor of the promotion of the toll station.

First, after seeking examples of toll stations in common use and analyzing the complicated situations, we enumerate the important elements which will quantify the performance of the station design. The performance is mainly constituted by throughput, accident rate and cost. Their weights are determined by principal component analysis.

Next, we applique Nagel-Schreckenberg (NS) model and then develop a function model to simulate the process for vehicles to pull out and every decision made by observing the surroundings. Meanwhile we can model the accident probability as well. In which way the quantity of vehicles via the toll station in unit time is obtained. And the cost is defined by the area the station takes and the number and proportions of tollbooths.

Then, according to this generalized adaptive model, we can adjust some parameters to reach a bigger throughput. And we recalculate accident rate and cost. By means of adjustments and experiments on MATLAB, a better solution with obvious advantage is designed. Its size, shape and merging pattern are detailed in our report.

Finally, with different parameters set, the condition of heavy traffic or light traffic is established, in which we compare the performance of our design with the common ones. The situation where more autonomous vehicles are added or the proportions of tollbooths are changed can also be tested in this model. The result turns out that our solution is an optimized design.

Keywords: Nagel-Schreckenberg (NS) model;

Contents

1 A letter to the New Jersey Turnpike Authority **2**

2 Introduction **2**

2.1 Statement of the problem 2

2.2 Other Assumptions 2

3 Analysis of the Problem **3**

4 Calculating and Simplifying the Model **4**

5 The Model Results **4**

6 Validating the Model **4**

7 Conclusions **4**

8 A Summary **4**

9 Evaluate of the Mode **4**

10 Strengths and weaknesses **4**

10.1 Strengths 4

Appendices **4**

Appendix A First appendix **4**

Appendix B Second appendix **5**

1 A letter to the New Jersey Turnpike Authority

New Jersey Turnpike Authority,

Our team has proposed an optimized toll station design allowing the increase of throughput and the decrease of cost and accident rate. And a new mathematical model is built in order to help evaluate the performances of designs.

More specifically, the performance model is developed after taking all important elements related to this problem into consideration, like number of lanes and tollbooths, proportions of tollbooths, varieties of vehicles, change of flows, every decision made by drivers to turn right or left and traffic conditions.

2 Introduction

2.1 Statement of the problem

The design of toll station is undoubtedly a kind of art as it is hard to find the balance among safety, capacity and cost facing different situations. But As a promotion of toll station is in demand, we applique mathematical methods and function models for optimize our design.

Here we list the elements that will influence the throughput of our toll station:

- Number of lanes and tollbooths L, B
- Total flow F_t
- Probability of large-scale automobiles, midium-sized vehicles and compact cars P_l, P_m, P_s
- Cell size C

Theorem 2.1.

Lemma 2.2.

Proof. The proof of theorem. □

2.2 Other Assumptions

-
-
-
-

Figure 1: aa

3 Analysis of the Problem

$$a^2 \tag{1}$$

$$\begin{pmatrix} *20ca_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \frac{Opposite}{Hypotenuse} \cos^{-1}\theta \arcsin\theta$$

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

$$p_j = \begin{cases} 0, & \text{if } j \text{ is odd} \\ r!\,(-1)^{j/2}, & \text{if } j \text{ is even} \end{cases}$$

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

$$\arcsin\theta = \oiiint_{\varphi} \lim_{x\to\infty} \frac{n!}{r!\,(n-r)!} \tag{1}$$

4 Calculating and Simplifying the Model

5 The Model Results

6 Validating the Model

7 Conclusions

8 A Summary

9 Evaluate of the Mode

10 Strengths and weaknesses

10.1 Strengths

- **Applies widely**
- **Improve the quality of the airport service**
-

Appendices

Appendix A First appendix

Here are simulation programmes we used in our model as follow.

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

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

```
//==========================================================================
// Name        : Sudoku.cpp
// Author      : wzlf11
// 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;
}
```