Q Value-based Dynamic Programming with Boltzmann Distribution for Traffic Management

YU, Shanqing

June 2011
Abstract

As the result of increased motorization and urbanization, modern metropolises with overcrowded traffics have been long suffering from traffic problems such as traffic congestions and traffic accidents. The frequently occurred traffic congestion, increasing the vehicles’ traveling time and air pollution, reduces the efficiency of transportation and brought a lot of troubles for the daily life.

Fortunately, as the development of Global Positioning System (GPS) and computer technology, the Intelligent Transportation Systems (ITS), which could collect the real-time information from the traffic systems and provide routes suggestions to the vehicles, have been proved quite effective on improving the efficiency of traffic systems and solving the problem of the traffic congestion, while ensuring traffic safety.

Meanwhile, how to do the traffic assignment and routes selection has attracted a greater attention recently as one of the most important parts of the vehicle navigation systems in ITS. The strategies in most of the traditional navigation systems, which inform the drivers with similar preferences of the same optimal route, may turn out to be a suboptimal solution in the overcrowded traffic system, since the most of the drivers may select the routes with the shortest traveling time as their optimal routes, which consequently causes some negative behavioral phenomena like concentration and overreaction. Therefore, how to make the optimal routes for all the
vehicles in the whole traffic system in a feasible and reasonable way arises as a challenging task for the centralized traffic scheduler.

Adopting multiple-paths algorithms, like the k-shortest paths algorithms and the disjoint path algorithms, is one of the solutions to split vehicles over several paths according to each driver preferences. Nevertheless, the multiple-path algorithms only succeed in searching multiple paths with certain constraints, but not consider how to assign the traffic to the selected paths.

In the recent decades, although many research is devoted to optimize the system performance of the traffic systems, there is still high interest in traffic assignment, particularly in the development of the approaches that can be deployed for large-scale real-time and planning applications.

In order to find a good approximation to the global optimum of the traffic system, a traffic management method, i.e., Q Value-based Dynamic Programming with Boltzmann Distribution has been proposed and systematically studied in this thesis. Unlike the multiple-paths algorithms, the proposed method not only calculates multiple paths, but also indicates which vehicles should be assigned to which routes in order to approximate the global optimum. In addition, the research in this thesis has proved that the proposed method could be applied to the real world large scale road networks with time-varying traffic information. It is also proved that the proposed method performs well on reducing the risk of the occurrence of the traffic congestion, saving the traveling cost and is less affected by the frequency of the real time traffic information update.

In chapter 2, the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution, in which Q Value-based Dynamic Programming and
Boltzmann distribution are simply combined to minimize the total traveling time of all the vehicles considering the traffic volumes, is proposed and applied to a small scale road network based on a static traffic assignment model and a dynamic traffic assignment model, respectively. The overall idea of the proposed method in this chapter is to update the traveling time of each section iteratively according to its corresponding traffic volume, and continuously generate new routes by the quasi-Q Value-based Dynamic Programming with Boltzmann distribution. Finally, a good approximation to the global optimum routes for all the Origin-Destination pairs could be found when the total traveling time converges. The simulation results demonstrate that the proposed method could perform better in global perspective theoretically compared to the conventional shortest-path method, i.e., greedy method.

In chapter 3, Q Value-based Dynamic Programming with Boltzmann Distribution, which is derived from the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution, is described and applied to a large scale road network, i.e., the Kitakyushu City road network, based on the static traffic assignment model. In the static traffic assignment model, the fixed traffic volume is given for each Origin-Destination pair and assigned to the route candidates, which are generated depending on the expected traveling time and the probability calculated by Q Value-based Dynamic Programming with Boltzmann Distribution. The traveling time of each section in the road network is evaluated by the well known Bureau of Public Roads (BPR) volume-delay function in each iteration for converging the traffic volume of the routes. The simulation results in this chapter reveal the efficiency of Q Value-based Dynamic Programming with Boltzmann Distribution and its feasibility to be applied to the large scale road
networks.

In chapter 4, Q Value-based Dynamic Programming with Boltzmann Distribution has been developed for a dynamic traffic management model by adding two temperature parameter control strategies, i.e., Network Method and Intersection Method, in order to reduce the traffic congestion in dynamic traffic systems with the time-varying traffic information. Network Method uses the same temperature parameter for the whole road network according to the global traffic situations, while Intersection Method adopts different temperature parameter for each intersection based on the traffic situations of sections connected to the intersection. The proposed dynamic traffic management model has been evaluated in a small size microscopic simulator in this chapter. The simulation results revealed that it is important to use the temperature parameter control in dynamic traffic systems and the proposed traffic management method can improve the system performance comparing with the conventional greedy method.

In chapter 5, a further advanced dynamic traffic management model is finally proposed and applied to the large scale microscopic simulator SOUND/4U based on the real world road network of Kurosaki, Kitakyushu in Japan. All the vehicles in the simulator follow the direction from the route guidance of the dynamic traffic management model, in which the extended Q Value-based Dynamic Programming with Boltzmann Distribution and the time-varying traffic information are used to generate the routes from the origins to destinations. The simulation results show that the proposed Q Value-based Dynamic Programming with Boltzmann Distribution could reduce the traffic congestion and improve the efficiency of the whole traffic system effectively compared with the greedy method in the real world road network.
In chapter 6, after giving the objective and motivation of each research in this thesis, some conclusions about the Q Value-based Dynamic Programming with Boltzmann Distribution and its applications are described based on the simulation results.
# Contents

Abstract ........................................................................................................... 1

Contents ........................................................................................................ 6

List of Figures ............................................................................................... 10

List of Tables .................................................................................................. 13

1 Introduction .................................................................................................. 1
   1.1 Background ............................................................................................ 1
   1.2 Traffic assignment .................................................................................. 3
   1.3 Dynamic Programming ......................................................................... 5
   1.4 Contents of this Research ...................................................................... 5
      1.4.1 Motivations .................................................................................... 5
      1.4.2 Research Topics ............................................................................ 6

2 Quasi-Q Value-based Dynamic Programming with Boltzmann Distribution ... 9
   2.1 Introduction ........................................................................................... 9
   2.2 Q Value-Based Dynamic Programming ............................................... 10
   2.3 Quasi-Q value-based Dynamic Programming with Boltzmann Distribution .................................................................................. 12
## CONTENTS

2.3.1 Generation of Route Candidates ........................................... 13
2.3.2 Evaluation of Route Candidates ............................................ 15
2.3.3 Traveling Time Updating .................................................... 15

2.4 Simulation in Static System .................................................. 18
2.4.1 Simulation Conditions ...................................................... 18
2.4.2 Results and Discussions .................................................. 19

2.5 Simulation in Dynamic System .............................................. 28
2.5.1 Traffic Flow Model ......................................................... 28
2.5.2 Simulation Conditions ...................................................... 30
2.5.3 Results and Discussions .................................................. 31

2.6 Conclusion ................................................................. 35

3 Q value-based Dynamic Programming with Boltzmann Distribution in Large Scale Road Network .................................................. 36

3.1 Introduction ................................................................. 36
3.2 BPR Volume-delay Function .................................................. 37
3.3 Q value-based Dynamic Programming with Boltzmann Distribution .................................................. 38
3.4 Description of the proposed method ........................................ 40
3.4.1 Temperature Parameter .................................................... 42
3.4.2 Generate Route Candidates .............................................. 45
3.4.3 Traffic Assignment .......................................................... 46

3.5 Simulation ................................................................. 47
3.5.1 Simulation Environment .................................................... 47
3.5.2 Analysis on $\theta$ and $\tau_0$ ............................................... 48
3.5.3 Analysis on the Number of Candidate Routes ....................... 51
3.5.4 Comparison of the Proposed Method and the Shortest Path Algorithm .................................................. 51
3.5.5 Computation time of the proposed method ......................... 55
3.6 Conclusion ................................................................. 56

4 Dynamic Traffic Management using Temperature Parameter Control in Q value-based Dynamic Programming with Boltzmann Distribution 58
4.1 Introduction ............................................................ 58
4.2 Structure of the Dynamic Traffic Management Model .............. 59
4.3 Network Method ....................................................... 61
4.4 Intersection Method .................................................. 61
4.5 Simulation .............................................................. 63
  4.5.1 Traffic Simulator ................................................. 63
  4.5.2 Temperature Parameter Analysis of Network Method .......... 67
  4.5.3 Temperature Parameter Analysis of Intersection Method ....... 68
  4.5.4 Comparison of Various Temperature Control Strategies ....... 68
4.6 Conclusion .............................................................. 70

5 An application of Q value-based Dynamic Programming with Boltzmann Distribution to real world road networks 76
5.1 Introduction ............................................................ 76
5.2 Dynamic Traffic Management Model ................................ 77
  5.2.1 Real-time traffic data collector ................................ 78
  5.2.2 Route guidance module ....................................... 78
  5.2.3 System evaluation module ..................................... 85
5.3 Simulation .............................................................. 86
  5.3.1 Simulation Condition ............................................ 86
  5.3.2 Temperature Parameter Analysis ................................ 89
  5.3.3 Comparison between the proposed method and the greedy method 90
5.4 Conclusion .............................................................. 94

6 Conclusions ............................................................... 95
List of Figures

2.1 Flow Chart of Quasi-Q value-based Dynamic Programming with Boltzmann Distribution ........................................ 12
2.2 Route candidates .......................................................... 13
2.3 A simple road network ................................................... 14
2.4 Traveling time function .................................................... 16
2.5 Simulation network (time unit) ......................................... 18
2.6 Total traveling time considering the traffic volumes ............. 19
2.7 Routes by Greedy strategy in Experiment 1 ....................... 20
2.8 Routes by Boltzmann strategy in Experiment 1 ................... 21
2.9 Total traveling time considering the traffic volumes ............. 23
2.10 Routes by Greedy strategy in Experiment 2 ....................... 24
2.11 Routes by Boltzmann strategy in Experiment 2 ................. 25
2.12 Comparison of different $\rho$ parameters ............................ 26
2.13 Comparison of different temperature parameters ............... 27
2.14 Flow Chart of Dynamic System ...................................... 29
2.15 Dynamic Simulation (case 1) .......................................... 32
2.16 Dynamic Simulation (case 2) .......................................... 33
2.17 Dynamic Simulation (case 3) .......................................... 34
3.1 Flow chart of the traffic assignment strategy ...................... 41
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>A simple road network</td>
<td>42</td>
</tr>
<tr>
<td>3.3</td>
<td>The optimal routes from each intersection to the destination</td>
<td>43</td>
</tr>
<tr>
<td>3.4</td>
<td>The Q values and probabilities of each section when using different</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>temperature parameter $\tau_i^{(n)}$</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>Kitakyushu City’s map</td>
<td>47</td>
</tr>
<tr>
<td>3.6</td>
<td>Analysis of $\theta$ with different $\tau_0$</td>
<td>49</td>
</tr>
<tr>
<td>3.7</td>
<td>Analysis of $\tau_0$ with different traffic conditions</td>
<td>49</td>
</tr>
<tr>
<td>3.8</td>
<td>Comparison of the average traveling time using different number of</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>candidate routes</td>
<td></td>
</tr>
<tr>
<td>3.9</td>
<td>Comparison between the the proposed method and shortest path algorithm</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>under different traffic conditions</td>
<td></td>
</tr>
<tr>
<td>3.10</td>
<td>The average number of the sections in five congestion risk levels</td>
<td>54</td>
</tr>
<tr>
<td>4.1</td>
<td>Flow chart of the dynamic traffic management model</td>
<td>60</td>
</tr>
<tr>
<td>4.2</td>
<td>Network Method</td>
<td>62</td>
</tr>
<tr>
<td>4.3</td>
<td>Intersection Method</td>
<td>63</td>
</tr>
<tr>
<td>4.4</td>
<td>Road network for simulations</td>
<td>64</td>
</tr>
<tr>
<td>4.5</td>
<td>An example of intersections in the road network</td>
<td>65</td>
</tr>
<tr>
<td>4.6</td>
<td>Comparison of different temperature parameters</td>
<td>66</td>
</tr>
<tr>
<td>4.7</td>
<td>Comparison of different parameter setting in Intersection Method</td>
<td>69</td>
</tr>
<tr>
<td>4.8</td>
<td>Comparison of three methods under different information update interval</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>in Small traffic situation</td>
<td></td>
</tr>
<tr>
<td>4.9</td>
<td>Comparison of three methods under different information update interval</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>in Middle traffic situation</td>
<td></td>
</tr>
<tr>
<td>4.10</td>
<td>Comparison of three methods under different information update interval</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>in High traffic situation</td>
<td></td>
</tr>
<tr>
<td>4.11</td>
<td>Comparison of different strategies</td>
<td>74</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Structure of dynamic traffic management model</td>
<td>77</td>
</tr>
<tr>
<td>5.2</td>
<td>Example of logistic network construction</td>
<td>79</td>
</tr>
<tr>
<td>5.3</td>
<td>A simple network</td>
<td>81</td>
</tr>
<tr>
<td>5.4</td>
<td>The conventional Q value-based Dynamic Programming with Boltzmann Distribution</td>
<td>82</td>
</tr>
<tr>
<td>5.5</td>
<td>The new Q value-based Dynamic Programming with Boltzmann Distribution</td>
<td>83</td>
</tr>
<tr>
<td>5.6</td>
<td>Simulation road network</td>
<td>87</td>
</tr>
<tr>
<td>5.7</td>
<td>Comparison of temperature parameters in different traffic situations</td>
<td>88</td>
</tr>
<tr>
<td>5.8</td>
<td>Average $\lambda_{od}$ of different OD sets in Experiment 1</td>
<td>90</td>
</tr>
<tr>
<td>5.9</td>
<td>Average $\lambda_{od}$ of different OD sets in Experiment 2</td>
<td>91</td>
</tr>
<tr>
<td>5.10</td>
<td>Comparisons in Experiment 1</td>
<td>92</td>
</tr>
<tr>
<td>5.11</td>
<td>Comparisons in Experiment 2</td>
<td>93</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Parameter setting of static simulations</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Set of OD pairs</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Traffic volumes (vehicles/time unit) of OD pairs in Experiment 1</td>
<td>22</td>
</tr>
<tr>
<td>2.4</td>
<td>Total traveling time of static systems</td>
<td>28</td>
</tr>
<tr>
<td>2.5</td>
<td>Parameter setting of dynamic simulations</td>
<td>31</td>
</tr>
<tr>
<td>3.1</td>
<td>Average and deviation of $E$ with different combinations of $\theta$ and $\tau_0$</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Parameter setting under different traffic conditions</td>
<td>52</td>
</tr>
<tr>
<td>3.3</td>
<td>Five congestion risk levels</td>
<td>52</td>
</tr>
<tr>
<td>3.4</td>
<td>The number of sections with very small traffic volumes</td>
<td>55</td>
</tr>
<tr>
<td>3.5</td>
<td>The size of the maps</td>
<td>55</td>
</tr>
<tr>
<td>3.6</td>
<td>The computation time(ms) of different maps with changing $\tau_0$</td>
<td>56</td>
</tr>
<tr>
<td>3.7</td>
<td>The computation time(ms) of shortest path algorithms</td>
<td>56</td>
</tr>
<tr>
<td>4.1</td>
<td>Signal control of simulator</td>
<td>64</td>
</tr>
<tr>
<td>4.2</td>
<td>Traffic situations in simulations</td>
<td>67</td>
</tr>
<tr>
<td>4.3</td>
<td>Parameter setting for simulations</td>
<td>68</td>
</tr>
<tr>
<td>4.4</td>
<td>Parameter setting for Network Method</td>
<td>68</td>
</tr>
<tr>
<td>5.1</td>
<td>Data of road network</td>
<td>86</td>
</tr>
<tr>
<td>5.2</td>
<td>OD sets</td>
<td>90</td>
</tr>
</tbody>
</table>
LIST OF TABLES

5.3 Comparison between the proposed method and the greedy method \ldots 91
Chapter 1

Introduction

1.1 Background

As the result of increased motorization and urbanization, modern metropolises with overcrowded traffics have been long suffering from traffic problems such as traffic congestions and traffic accidents. The frequently occurred traffic congestion, increasing the vehicles’ traveling time and air pollution, reduces the efficiency of transportation and brought a lot of troubles for the daily life. Terrible traffic congestion even could ruin people’s emotion.

Fortunately, as the development of Global Positioning System (GPS) and computer technology, the Intelligent Transportation Systems (ITS), such as VICS (Vehicle Information and Communication System) [1][2] in Japan and TravTek [3] in America, could collect real-time information from the traffic systems, provide routes suggestions to the vehicles and have been proved quite effective on improving the efficiency of traffic systems and solving the problem of the traffic congestion, while ensuring traffic safety [4][5][6][7][8].

Meanwhile, how to do the traffic assignment and routes selection, as one of the most important parts of the vehicle navigation systems in ITS, has attracted a greater
attention, recently. However, strategies in the most of the traditional navigation systems, which informs the drivers with similar preferences of the same optimal route, may turn out to be a suboptimal solution in the overcrowded traffic system, since the most of the drivers may select the routes with shortest traveling time as their optimal routes and consequently causes some negative behavioral phenomena like concentration and overreaction [9][10]. Concentration means that the traffic volume centers on the same optimal route which consequently causes the traffic jam, and overreaction means that the vehicles on the optimal route would turn to another route after the update of the information, which results in the unexpected low traffic volume on the previous optimal route. Therefore, how to make the optimal routes for all the vehicles in the whole traffic system in a feasible and reasonable way arises as a challenging task for the centralized traffic scheduler.

Developing multiple-path algorithms, like the k-shortest path algorithms [11][12][13] and the disjoint path algorithms [14][15], is one of the solutions to split vehicles over several paths according to each driver preferences. Nevertheless, the multiple-path algorithms only succeed in searching multiple paths with certain constraints but not consider how to assign the traffic to the selected paths.

Researchers in [16] calculates k shortest paths for the drivers every operational time interval. However, it just applied to a very simple road network and they just equally distribute the traffic volume to the generated shortest paths. In addition, in order to get the global optimal routes, an additional path checking procedure has to be performed to select the paths with acceptable constraints for the drivers. [17] proposed a Pretrip multipath planning method, in which some partly disjoint alternative paths are calculated offline to render the system less reliant on real-time traffic information and consequently save the computation time during the trip. However, this method only provides alternative paths for the drivers when the original route is not available and it is not aiming at reducing the traffic congestion and improving the traffic system.
performance in global perspective.

Although many research is devoted to optimize the system performance of the traffic systems, there is still heightened interest in traffic assignment, particularly in the development of approaches that can be deployed for large-scale real-time and planning applications [18].

1.2 Traffic assignment

Traffic assignment including travel choice principle and travel cost function, is long recognized as a key component for network planning and transport policy evaluations.

Numerous formulations and solutions have been proposed for the purpose of reducing the traffic congestion. Dafermos applied the Frank-Wolfe algorithm, which include the well known Bureau of Public Roads (BPR) volume-delay function, to deal with the traffic equilibrium problem in 1968s [19][20]. Wardrop proposed the famous Wardrop equilibrium rule on how to assign the traffic to paths and links in 1952s[21]. And there are lots of researchers developed their system based on the equilibrium equation [22][23][24].

As the intelligent traffic system develops, the dynamic traffic assignment (DTA) has attracted more attention recently, since the static traffic assignment [25][26][27] is not describing the dynamic traffic flow of the network[28][29].

 Dynamic traffic assignment, though still in a state of flux, has evolved substantially since the semina work of [30] [31]. DTA refers to a broad spectrum of problems, each corresponding to different sets of decision variables, underlying behavioral and
system assumptions. One common feature of these models is that they depart from the standard static assignment assumptions to deal with time-varying flows[18].

There are four categories of methods that arise in the study of DTA: mathematical programming, optimal control, variational inequality, and simulation-based.

Mathematical programming DTA models, whose first attempt are represented by Merchant and Nemhauser [30][31], formulate the problem in a discretized time-setting. The formulation was limited to the deterministic, fixed-demand, single-destination, single-commodity and system optimal case. The most famous phenomenon related to the system optimal mathematical programming models that impinges on traffic realism is the FIFO [32][33] and the ”holding-back” of vehicles on links [34]. The traffic assignment models in this thesis are also based on the FIFO.

In optimal control theory DTA formulations, the OD traffic demand are assumed as known and time-varying. Friesz in [35] discussed optimal control formulation for both the system optimal and user equilibrium (UE) objectives for the single destination case, while Ran and Shimazaki use the optimal control approach to develop a system optimal model for an urban transportation network with multiple ODs [36][37]. However, they applied their method to the unrealistic model without considering reasonable mathematical theory, such as FIFO.

Variational inequality provides great analytical flexibility and convenience for many kinds of problems in the DTA such as optimization, fixed point and complementarity. However, the computationally intensive approaches in this field make it difficult to be applied to the real-time deployment.

As the development of computer technology, Simulation-based DTA models, which use a traffic simulator to replicate the complex traffic flow dynamics, critical for developing meaningful operational strategies for real-time deployment, become quite popular.

In addition to the use of a simulator in a descriptive mode to determine the traffic
flow propagation, most existing simulation-based models also use it as a part of the search process to determine the optimal solution [38][39]. In this thesis, most of the proposed methods are applied to a simulation-based traffic assignment model.

1.3 Dynamic Programming

Dynamic programming is a method for efficiently solving a broad range of search and optimization problems which exhibit the characteristics of overlapping subproblems and optimal substructure.

The Q Value-based Dynamic Programming [40] is based on the principle of dynamic programming. Different with the well-known Bellman equation [41], the Q Value-based Dynamic Programming not only searches for the optimal traveling time but also tell the best next intersection to move explicitly.

Meanwhile, the Q Value-based Dynamic Programming with Boltzmann Distribution, in which the next selected section is unknown and probability depended, is developed based on the principle of stochastic dynamic programming [42]. The main difference between the Q Value-based Dynamic Programming with Boltzmann Distribution and the stochastic dynamic programming methods is that instead of unknown rewards, the traveling time of each section in the current time can be estimated.

1.4 Contents of this Research

1.4.1 Motivations

In order to avoid the disadvantages of the traditional greedy guiding strategy and find a good approximation to the global optimum of the traffic system, a traffic management method, i.e., Q Value-based Dynamic Programming with Boltzmann Distribution, which could be applied to the real world large scale road networks with
time-varying traffic information, has been proposed and systematically studied in this thesis.

Unlike the multiple-paths algorithms, the proposed method not only calculates multiple paths, but also shows which vehicles should be assigned to which routes in order to approximate the global optimum.

The research is carried out by gradually studying the performance of the proposed method from simple static simulation to large scale real-time simulation. In addition, the simulation results revealed the feasibility of the proposed method to be applied to the real world large scale road networks with time-varying traffic information and also proved that the proposed method performs well on reducing the risk of the occurrence of the traffic congestion, saving the traveling cost and is less affected by the frequency of the real time traffic information collection.

1.4.2 Research Topics

In this thesis, there are four topics discussed based on the former mentioned motivations.

In chapter 2, the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution, in which Q Value-based Dynamic Programming and Boltzmann distribution are simply combined to minimize the total traveling time of all the vehicles considering the traffic volumes, is proposed and applied to a small scale road network based on a static traffic assignment model and a dynamic traffic assignment model, respectively. The overall idea of the proposed method in this chapter is to update the traveling time of each section iteratively according to its corresponding traffic volume, and continuously generate new routes by the quasi-Q Value-based Dynamic Programming with Boltzmann distribution. Finally, a good approximation to the global optimum routes for all the Origin-Destination pairs could be found when the total traveling time converges. The simulation results demonstrate that the proposed method could perform
better in global perspective theoretically compared to the conventional shortest-path method, i.e., greedy method.

In chapter 3, the Q Value-based Dynamic Programming with Boltzmann Distribution, which is derived from the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution, is described and applied to a large scale road network, i.e., the Kitakyushu City road network, based on the static traffic assignment model. In the static traffic assignment model, the fixed traffic volume is given for each Origin-Destination pair and assigned to the route candidates, which are generated depending on the expected traveling time and the probability calculated by Q Value-based Dynamic Programming with Boltzmann Distribution. The traveling time of each section in the road network is evaluated by the well known Bureau of Public Roads (BPR) volume-delay function in each iteration for converging the traffic volume of the routes. The simulation results in this chapter reveal the efficiency of Q Value-based Dynamic Programming with Boltzmann Distribution and its feasibility of to be applied to the large scale road networks.

In chapter 4, Q Value-based Dynamic Programming with Boltzmann Distribution has been developed for a dynamic traffic management model by adding two temperature parameter control strategies, i.e., Network Method and Intersection Method, in order to reduce the traffic congestion in dynamic traffic systems with the time-varying traffic information. Network Method uses the same temperature parameter for the whole road network according to the global traffic situations, while Intersection Method adopts the different temperature parameter for each intersection based on the traffic situations of sections connected to the intersection. The proposed dynamic traffic management model has been evaluated in a small size microscopic simulator in this chapter. The simulation results revealed that it is important to use the temperature parameter control in dynamic traffic systems and the proposed traffic management method can improve the system performance comparing with the conventional greedy
method.

In chapter 5, a further advanced dynamic traffic management model is finally proposed and applied to the large scale microscopic simulator SOUND/4U based on the real world road network of Kurosaki, Kitakyushu in Japan. All the vehicles in the simulator follow the direction from the route guidance of the dynamic traffic management model, in which the extended Q Value-based Dynamic Programming with Boltzmann Distribution and the time-varying traffic information are used to generate the routes from the origins to destinations. The simulation results show that the proposed Q Value-based Dynamic Programming with Boltzmann Distribution could reduce the traffic congestion and improve the efficiency of the whole traffic system effectively compared with the greedy method in the real world road network.

In chapter 6, after giving the objective and motivation of each research in this thesis, some conclusions about the Q Value-based Dynamic Programming with Boltzmann Distribution and its applications are described based on the simulation results.
Chapter 2

Quasi-Q Value-based Dynamic Programming with Boltzmann Distribution

2.1 Introduction

In order to avoid the disadvantages of the traditional greedy guiding strategy and improve the efficiency of traffic systems in global perspective, a heuristic routing algorithm, i.e., the quasi-Q value-based Dynamic Programming with Boltzmann Distribution [43], in which Q value-based Dynamic Programming [40] and Boltzmann distribution [44] are simply combined to minimize the total traveling time of all the vehicles considering the traffic volumes, has been proposed and evaluated in the small scale road network both in the static traffic system and dynamic traffic system in this chapter.

Instead of other conventional shortest-path search algorithms [45][46][47], Q value-based Dynamic Programming is adopted to calculate the optimal traveling time to each destination from every intersection of the road networks, because it has the following
two distinguished advantages over others:

- The conventional methods only search for the optimal route, but can’t search for the second best or the third best one, while Q value-based Dynamic Programming can tell us which intersection is the best one or second best one to move in the next.

- Another advantage of Q value-based Dynamic Programming is less computationally intensive and easy to search for alternative optimal routes when the traveling time of the road sections changes. Most of the optimal route search algorithms would delete the original solution and recompute everything from scratch, when the traveling time of the road networks changes, while Q value-based Dynamic Programming exploits the available recent information and updates the solution with a minimum number of computations. Actually, in the case of searching the optimal route in dynamic traffic systems where the traveling time of sections are changing frequently, the optimal route searching algorithm like Q value-based Dynamic Programming is what exactly we need.

In section 2.2, the outline of Q value-based Dynamic Programming is reviewed, while the details of the quasi-Q value-based Dynamic Programming with Boltzmann Distribution are described in section 2.3. Section 2.4 shows the simulations in the static traffic system and section 2.5 presents the simulations in the dynamic traffic system. Section 2.6 is devoted to conclude the contribution of this chapter.

2.2 Q Value-Based Dynamic Programming

Q value-based Dynamic Programming is an iterative Q value updating algorithm aiming to search for the optimal route and its optimal traveling time for a given Origin-Destination (OD) pair of road networks. The Q value—\( Q_d(i, j) \), which is defined as
the minimum traveling time to destination \( d \) when a vehicle bound for destination \( d \) moves to intersection \( j \) at intersection \( i \), is calculated iteratively based on the following equation.

\[
Q_d(i, j) \leftarrow t_{ij} + \min_{k \in A(j)} Q_d(j, k),
\]

(2.1)

where,

\( i, j \in I \) : suffixes of intersections and their set
\( d \in D \) : suffix of destinations and its set
\( t_{ij} \) : traveling time from intersection \( i \) to intersection \( j \)
\( A(i) \) : set of suffixes of intersections moving directly from intersection \( i \)

Usually, the update of \( Q \) values starts from destination intersections. \( Q \) values for all the pairs of adjacent intersections are initialized as follows.

\[
Q^{(0)}_d(d, j) = 0, \quad d \in D, \ j \in A(d)
\]

(2.2)

\[
Q^{(0)}_d(i, d) = t_{id}, \quad i \in B(d)
\]

(2.3)

\[
Q^{(0)}_d(i, j) = 0, \quad i \in I - \{d\} - B(d), \ j \in A(i)
\]

(2.4)

where,

\( B(i) \) : set of suffixes of intersections moving directly to intersection \( i \)

\( Q \) values of \( n^{th} \) iteration, i.e., \( Q^{(n)}_d(i, j) \) is updated based on the following equations.

\[
Q^{(n)}_d(d, j) = 0, \quad d \in D, \ j \in A(d)
\]

(2.5)
2.3 Quasi-Q value-based Dynamic Programming with Boltzmann Distribution

Fig. 2.1 shows the flow chart of the proposed quasi-Q value-based Dynamic Programming with Boltzmann Distribution [48][49]. In the following subsections, the
2.3.1 Generation of Route Candidates

In order to search for the global optimal route considering the traffic volumes, some candidates of the optimal routes shown in Fig. 2.2 should be generated in the proposed method. When generating route candidates, not only the shortest-path, but also the other routes to the destination have a chance to be selected according to the probability calculated by Q values. The following equation explains how to use Boltzmann distribution to calculate \( P_d(i, j) \), i.e., the probability that the vehicle bound for destination \( d \) moves to intersection \( j \) at intersection \( i \).

\[
P_d(i, j) = \frac{e^{-\frac{Q_d(i, j)}{\tau}}}{\sum_{j \in A(i)} e^{-\frac{Q_d(i, j)}{\tau}}},
\]

where \( G \) is the set of suffixes of candidates, \( R^g \) is the candidate of the optimal route, \( R^g_{|D|} \) is the candidate of the optimal route to destination \( d \), and \( R^g_{|G|} \) is the candidate of the optimal route to destination \( g \).
where,

$\tau$: a parameter called temperature

![Figure 2.3: A simple road network](image)

Take the road network in Fig. 2.3 describing that $Q_d(o, a) = 9$; $Q_d(o, b) = 5$; $Q_d(o, c) = 11$. Suppose $\tau = 2$, then, $P_d(o, a) = 0.1142$; $P_d(o, b) = 0.8438$; $P_d(o, c) = 0.0420$. The vehicle going to destination $d$ from origin $o$ will select the shortest path $(o, b)$ in the traditional method. However, in the new proposed method, the vehicle has 11.42 percent probability to select $(o, a)$, 84.38 percent to choose $(o, b)$ and 4.20 percent to pick up $(o, c)$.

Basically, the probability is likely to be inversely proportional to Q values. However, the parameter "temperature" in Boltzmann distribution controls the impact of Q values to the probability. When the "temperature" is very high, Boltzmann distribution is identical to random distribution in which each intersection has equal opportunities to be selected. On the contrary, when the "temperature" approaches 0, only the optimal route is available, just like Greedy strategy.

In the proposed method, the "temperature" starts from a high value in which the route determination process behaves more like an exploration strategy, and gradually gets lower over iterations in which the route determination process works like an ex-
exploitation mechanism.

2.3.2 Evaluation of Route Candidates

Among the route candidates generated by Boltzmann distribution, the global optimal route \( \bar{g} \) considering traffic volumes is selected by the following equation.

\[
\bar{g} = \arg \min_{g \in G} \left\{ \sum_{o \in O} \sum_{d \in D} \sum_{(i,j) \in R_{od}(g)} v_{od} t_{ij} \right\},
\]

(2.9)

where,

\( o \in O \): suffix of origins and its set  
\( R_{od}(g) \): \( g^{th} \) candidate of the optimal route from origin \( o \) to destination \( d \)  
\( v_{od} \): traffic volume from origin \( o \) to destination \( d \)  
\( t_{ij} \): traveling time of section \( s_{ij} \) when using \( R_{od}(g) \)  
\( s_{ij} \): section from intersection \( i \) to intersection \( j \)

The sum of the product of the traffic volume and traveling time on each section on route \( R_{od}(g) \) over all origin and destination pairs is minimized with respect to \( g \in G \) in order to obtain the global optimal route \( R_{od}(\bar{g}) \).

2.3.3 Traveling Time Updating

<Traveling Time Function>

In practice, the traveling time of one specific route section could be influenced by many real factors and uncertainties, one of which is the change of the traffic volume. In this chapter, we introduce the traveling time function \( f(v_{ij}(\bar{g})) \) which is initialized
to the distance of the section, and constantly changing with the movement of cars on the section as shown in Fig. 2.4. When the traffic volume on the section exceeds the capacity, the traveling time increases exponentially according to the follow equation.

\[ f(v_{ij}(\overline{g})) = \alpha d_{ij} e^{\beta v_{ij}(\overline{g})}, \] (2.10)

\[ v_{ij}(\overline{g}) = \sum_{o \in O} \sum_{d \in D} \begin{cases} v^{od}, & \text{if } s_{ij} \in R_{od}(\overline{g}) \\ 0, & \text{if } s_{ij} \notin R_{od}(\overline{g}) \end{cases} \] (2.11)

where,

\( \alpha, \beta \) : constant

\( d_{ij} \) : distance of section \( s_{ij} \)

\( v_{ij}(\overline{g}) \) : traffic volume on the section \( s_{ij} \) when using \( R_{od}(\overline{g}) \)
Table 2.1: Parameter setting of static simulations

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of candidates</td>
<td>20</td>
</tr>
<tr>
<td>Number of Iterations</td>
<td>800</td>
</tr>
<tr>
<td>$\alpha$ coefficient from traffic volume to traveling time</td>
<td>1</td>
</tr>
<tr>
<td>$\beta$ coefficient from traffic volume to traveling time</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Table 2.2: Set of OD pairs

<table>
<thead>
<tr>
<th>OD Set</th>
<th>Number of Pairs (vehicles/time unit)</th>
<th>Set of Origin Number</th>
<th>Set of Destination Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>$OD1$</td>
<td>16</td>
<td>$N1, N2, N3, N4$</td>
<td>$N71, N72, N73, N74$</td>
</tr>
<tr>
<td>$OD2$</td>
<td>49</td>
<td>$N1, N2, N3, N4, N5, N6, N7$</td>
<td>$N71, N72, N73, N74, N75, N76, N77$</td>
</tr>
</tbody>
</table>

<Traveling Time Smoothing>

In order to update the traveling time gradually and converge at the end of the iterations, the traveling time of each section is updated every iteration according to the value of $f(v_{ij}(\bar{g}))$ by the following smoothing equation.

$$t_{ij} = \rho t_{ij} + (1 - \rho)f(v_{ij}(\bar{g})), \quad 0 \leq \rho \leq 1,$$

(2.12)

The parameter $\rho$ equals 0 at the beginning of the iterations and increases gradually until it equals 1. Obviously, the traveling time of the section $t_{ij}$ will converge when $\rho$ reaches 1.
2.4 Simulation in Static System

2.4.1 Simulation Conditions

Firstly, we have simulated the proposed quasi-Q value-based Dynamic Programming with Boltzmann Distribution in the static system where the volume from the origin to destination is supposed to be constant. The simulation has been done to the road network with 77 intersections shown in Fig. 2.5.

The x in (x,y) on the horizontal roads represents the traveling time from left intersection to right intersection and y means right to left, while, (x,y) value on the vertical roads represents the traveling time from up to down and down to up. The traveling time for the road sections are initialized randomly in the range of 1 to 10 time units. Each
road section is bidirectional, but the traveling time of going in the reverse direction could be different.

Table 2.1 shows the parameter setting in the static simulations. Two sets of OD pairs with different traffic volumes shown in Table 2.2 have been tested in our simulation.

2.4.2 Results and Discussions

<Experiment 1 using OD1>

The total traveling time considering the traffic volumes VT which is calculated by Eq.(2.13) is used to evaluate the efficiency of the methods as shown in Fig.2.6. The
Figure 2.7: Routes by Greedy strategy in Experiment 1
Figure 2.8: Routes by Boltzman strategy in Experiment 1
Table 2.3: Traffic volumes (vehicles/time unit) of OD pairs in Experiment 1

<table>
<thead>
<tr>
<th>O</th>
<th>N71</th>
<th>N72</th>
<th>N73</th>
<th>N74</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>15</td>
<td>35</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>N2</td>
<td>30</td>
<td>15</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>N3</td>
<td>15</td>
<td>30</td>
<td>35</td>
<td>20</td>
</tr>
<tr>
<td>N4</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
</tr>
</tbody>
</table>

input traffic volumes of OD1 shows in Table 2.3.

\[ VT = \sum_{o \in O} \sum_{d \in D} \sum_{(i,j) \in R_{o,d}(E)} v^{od}_{ij}, \]  

(2.13)

With Boltzmann strategy, the curve in Fig.2.6 shows that the total traveling time considering the traffic volumes \( VT \) gradually converges through iterations, while \( VT \) with Greedy strategy does not converge. In this case, we define the iteration number as 800. However, 800 iterations occupy 7735 machine times to calculate, which is only a few seconds. It is quite acceptable, and also the total evaluation index \( VT \) of the Boltzmann strategy is definitely much better than the Greedy strategy.

Fig.2.7(a)–Fig.2.7(d) show how the routes change during the iterations when using Greedy strategy. The number shows the volume on the sections. After the 4\(^{th}\) iteration, the routes alternated between Fig.2.7(c) and Fig.2.7(d), and the corresponding total evaluation index \( VT \) also alternated between 48206.10 and 31718.40 vehicles as shown in Fig.2.6. It shows that vehicles go through the same optimal route, and cause a traffic jam as shown in Fig.2.7(c), and then turn to other roads in Fig.2.7(d) expecting the low traffic volume on the optimal route in Fig.2.7(c).

Fig.2.8(a)–Fig.2.8(c) demonstrate how the routes change as the iteration goes on when using the Boltzmann strategy. The route solution shown in Fig.2.8(c) is the one we found at the end of the iterations. They are considered as the global optimal routes.
Figure 2.9: Total traveling time considering the traffic volumes (Experiment 2, OD Set: OD2, Volume: 4 vehicles/time unit)

where the total evaluation index $VT$ equals 18991.60 vehicles.

In Fig.2.7-Fig.2.8, the gray arrows denote the routes and the number on the arrows or beside the arrows is the traffic volume on the sections. It is obvious that the average traffic volume on the sections using Boltzmann strategy is much smaller than using Greedy strategy.

**<Experiment 2 using OD1>**

Fig.2.9 shows the total evaluation index $VT$ using OD2 with the traffic volume from the origin to destination being equal to 4 vehicles/time unit.

It is found from Fig.2.9 that the routes obtained by the Greedy strategy alternated between Fig.2.10(a) and Fig.2.10(b) after the 11th iteration, and the corresponding $VT$ alternated between 10146.00 vehicles and 10695.20 vehicles.
(a) 10th iteration ($VT = 10146.00$ vehicles)  (b) 11th iteration ($VT = 10695.20$ vehicles)

Figure 2.10: Routes by Greedy strategy in Experiment 2

The curve by Boltzmann Optimal Route Method in Fig. 2.9 shows that $VT$ gradually converges through iterations. The routes shown in Fig. 2.11 are the global optimal routes found at the end of the iterations where $VT$ equals 8124.00 vehicles.

<Parameter Sensitivity Analysis>

As we described at the beginning of this chapter, the quasi-Q value-based Dynamic Programming with Boltzmann Distribution is a heuristic method trying to find a good approximation to the global optimum route by updating the traveling time of each route section and the optimal route, iteratively.

As to the traveling time updating, we use parameter $\rho$ to smooth the fluctuation of $f(v_{ij}(g))$ function. The principle is that $\rho$ should increase gradually from 0 to 1 through the iterations to ensure that the traveling time of each section could converge. Following this principle, the various changing patterns of $\rho$ were simulated and their results seem to have small differences according to Fig. 2.12. Therefore, we used $\rho = \rho_2$. 

24
For the route searching strategy, several changing patterns of parameter $\tau$ as shown in Eq.(2.8) were simulated shown Fig.2.13. The converged total evaluation index $VT$ had no big differences in four cases of $\tau$. So, $\tau_2$ is employed in the simulations.

<Discussions>

In order to further compare the quasi-Q value-based Dynamic Programming with Boltzmann Distribution under different traffic conditions from the previous simulations, we simulated these two methods over the OD sets shown in Table 2.4, in which ”Greedy.a” and ”Greedy.b” mean that the total evaluation index $VT$ of the routes found by Greedy strategy alternated between ”Greedy.a” and ”Greedy.b”, while ”Boltzmann” means the total evaluation index $VT$ of the final solution produced by Boltzmann strategy. Table 2.4 demonstrates that the traffics suffer greatly from overreaction and concentration by adopting Greedy strategy and as a result, Greedy strategy reduces the system performance especially in the case of heavy traffic volume.
Figure 2.12: Comparison of different $\rho$ parameters

(a) different $\rho$ parameters

(b) $\rho_1 = 1 - e^{-\frac{\text{iterations}}{200}}$

(c) $\rho_2 = 1 - e^{-\frac{\text{iterations}}{80}}$

(d) $\rho_3 = 1 - e^{-\frac{\text{iterations}}{40}}$

(e) $\rho_4 = 1 - e^{-\frac{\text{iterations}}{30}}$
(a) different $\tau$ parameters

(b) $\tau_1 = 0.1 + 10 \cdot e^{-\frac{\text{iterations}}{80}}$

(c) $\tau_2 = 0.1 + 10 \cdot e^{-\frac{\text{iterations}}{120}}$

(d) $\tau_3 = 0.1 + 10 \cdot e^{-\frac{\text{iterations}}{160}}$

(e) $\tau_4 = 0.1 + 10 \cdot e^{-\frac{\text{iterations}}{180}}$

Figure 2.13: Comparison of different temperature parameters
Table 2.4: Total traveling time of static systems

<table>
<thead>
<tr>
<th>OD</th>
<th>volume (vehicles/time unit)</th>
<th>Greedy.a (vehicles)</th>
<th>Greedy.b (vehicles)</th>
<th>Boltzmann (vehicles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OD1 6-16</td>
<td>7096.87</td>
<td>7808.05</td>
<td>5910.78</td>
</tr>
<tr>
<td>2</td>
<td>OD1 8-25</td>
<td>13017.00</td>
<td>17540.30</td>
<td>10655.10</td>
</tr>
<tr>
<td>3</td>
<td>OD1 15-40</td>
<td>48206.10</td>
<td>31718.40</td>
<td>18991.60</td>
</tr>
<tr>
<td>4</td>
<td>OD1 20-50</td>
<td>156710.00</td>
<td>70629.60</td>
<td>33882.30</td>
</tr>
<tr>
<td>5</td>
<td>OD2 4</td>
<td>10146.00</td>
<td>10695.20</td>
<td>8124.00</td>
</tr>
<tr>
<td>6</td>
<td>OD2 6</td>
<td>16646.70</td>
<td>15902.90</td>
<td>13267.40</td>
</tr>
<tr>
<td>7</td>
<td>OD2 8</td>
<td>30428.30</td>
<td>25544.50</td>
<td>19171.70</td>
</tr>
<tr>
<td>8</td>
<td>OD2 10</td>
<td>40767.00</td>
<td>49111.30</td>
<td>26115.10</td>
</tr>
</tbody>
</table>

In addition, the proposed quasi-Q value-based Dynamic Programming with Boltzmann Distribution decreases the total traveling time by distributing the traffic volumes to many sections of the roads. It is obvious that Boltzmann Method is much better than Greedy Method especially when Greedy Method suffers greatly from overreaction and concentration.

### 2.5 Simulation in Dynamic System

Fig. 2.14 is the flow chart of the dynamic systems using the quasi-Q value-based Dynamic Programming with Boltzmann Distribution. The quasi-Q value-based Dynamic Programming with Boltzmann Distribution is adopted periodically to calculate the optimal routes to guide the vehicles in which \( v_{od}(t) \) is used instead of \( v_{od} \) in the static system. How to simulate the traffic flow and update the traveling time of the sections will be explained in the following subsection.

#### 2.5.1 Traffic Flow Model
In the dynamic system, the traffic volume \( v_{jk}(t) \) on section \( s_{jk} \) at time \( t \) is supposed to be calculated by the following equations.

\[
v_{jk}(t) = \sum_{o \in O} \sum_{d \in D} v^{od}_{jk}(t), \quad j \in I, \ k \in A(j) \tag{2.14}
\]

\[
v^{od}_{jk}[t + t_{ij}(t)] = \begin{cases} 
\sum_{r \in R(i,j)} v^{od}_{ij}(t), & \text{if } s_{jk} \in R_{od}(\bar{g}) \\
0, & \text{if } s_{jk} \notin R_{od}(\bar{g})
\end{cases} \tag{2.15}
\]

\[
v^{od}_{oj}(t) = \begin{cases} 
v_{od}(t), & \text{if } s_{oj} \in R_{od}(\bar{g}) \\
0, & \text{if } s_{oj} \notin R_{od}(\bar{g})
\end{cases} \tag{2.16}
\]

\[ o \in O, \ d \in D, \ j \in I, \ k \in A(j), \]

\[ o \in O, \ d \in D, \ j \in A(o), \]
where,
\(v_{jk}(t)\) : traffic volume on section \(s_{jk}\) at time \(t\)
\(t_{ij}(t)\) : traveling time on section \(s_{jk}\) at time \(t\)
\(v^\text{od}_{jk}(t)\) : traffic volume from origin \(o\) to destination \(d\) on section \(s_{jk}\) at time \(t\)
\(v^\text{od}(t)\) : traffic volume for destination \(d\) at origin \(o\) at time \(t\)
\(R_{\text{opt}(d)}\) : the global optimal route from origin \(o\) to destination \(d\)

\(<\text{Traveling Time Updating}>\)

The traveling time of each section is firstly initialized to the distance of the section, and updated gradually according to the following equation.

\[ t_{ij}(t) = \sigma t_{ij}(t-1) + (1-\sigma)f(v_{ij}(t-1)), \quad 0 \leq \sigma \leq 1, \quad (2.17) \]

where,
\(\sigma\): parameter

The above equation seems similar to the traveling time smoothing equation in the static traffic system. However, these two equations have different meanings. The parameter \(\rho\) in the static traveling time smoothing equation increases gradually from 0 to 1 for the traveling time to converges at the end of the iterations. However, in the dynamic system, \(\sigma\) is constant just for smoothing the traveling time. The smaller \(\sigma\) is, the more the current traveling time is considered in the average operation.

\(2.5.2 \quad \text{Simulation Conditions}\)

The same road network as Fig.2.5 is used for the dynamic system simulations. The set of origin is \{\(N1, N2, N3, N4\}\), and the set of destination is \{\(N74, N75, N76, N77\)\}. 
Table 2.5: Parameter setting of dynamic simulations

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of candidates</td>
<td>G</td>
</tr>
<tr>
<td>Maximum time step</td>
<td>N</td>
</tr>
<tr>
<td>$\alpha$ coefficient from traffic volume to traveling time</td>
<td>1</td>
</tr>
<tr>
<td>$\beta$ coefficient from traffic volume to traveling time</td>
<td>0.005</td>
</tr>
<tr>
<td>$\sigma$ coefficient for smoothing the traveling time</td>
<td>0.5</td>
</tr>
<tr>
<td>Route recalculation time steps</td>
<td></td>
</tr>
</tbody>
</table>

The traffic volume $v_{od}(t)$ is inputted to the origin. Table 2.5 shows the parameter setting in the simulations.

The total traveling time considering the traffic volume, i.e., $DVT$ in the dynamic system is calculated by the following equation to evaluate the efficiency of the system.

$$DVT = \sum_{i \in I} \sum_{j \in A(i)} v_{ij}(t) t_{ij}(t),$$  \hspace{1cm} (2.18)

2.5.3 Results and Discussions

In this subsection, we describe the simulation results changing the traffic volume $v_{od}(t)$ as shown in Fig. 2.15–Fig. 2.17 to discuss the efficiency of the quasi-Q value-based Dynamic Programming with Boltzmann Distribution in the dynamic traffic systems. From them, we can see that in the dynamic system the Greedy strategy is also suffering overreaction since the total evaluation index $DVT$ changes suddenly sometimes, which would be studied in more details in our future work using more realistic traffic systems.

Except for the beginning of the simulations when the traffic volume in the road network is small, the $DVT$ of the Bolztmann strategy is always smaller than the $DVT$
Figure 2.15: Dynamic Simulation (case 1)
Figure 2.16: Dynamic Simulation (case 2)
(a) Averaged input volume

(b) Total traveling time $DVT$ considering the traffic volume in the dynamic system

Figure 2.17: Dynamic Simulation (case 3)
of the Greedy strategy. In conclusion, the quasi-Q value-based Dynamic Programming with Boltzmann Distribution reduces the total traveling cost effectively and performs much better than Greedy Method especially in the case of heavy traffic volumes.

2.6 Conclusion

In this chapter, a heuristic routing algorithm, i.e., the quasi-Q value-based Dynamic Programming with Boltzmann Distribution, where Q value-based Dynamic Programming and Boltzmann distribution are combined are described and analyzed. The proposed method is compared with the conventional shortest-path method, i.e., Greedy strategy in both static and approximates dynamic traffic systems. The simulation results demonstrate that the quasi-Q value-based Dynamic Programming with Boltzmann Distribution performs better than the conventional one in global perspective.

Although the simulation in this chapter is not realistic and the method still remain immature, the research in this chapter firstly proposed the concept of using Q value-based Dynamic Programming and Boltzmann distribution to do the global optimal routes selection and proved its efficiency and feasibility by comparing with the greedy method.
Chapter 3

Q value-based Dynamic Programming with Boltzmann Distribution in Large Scale Road Network

3.1 Introduction

In this chapter, the Q Value-based Dynamic Programming with Boltzmann Distribution, which is derived from the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution in chapter 2, is described and applied to a large scale road network, i.e., the Kitakyushu City road network, based on the static traffic assignment model.

The main idea of the proposed method is to calculate the expected traveling time for each Origin-Destination pair and the probability of selecting the next section, then to generate a considerable number of route candidates for the drivers based on the calculated probability.

In the static traffic assignment model, the fixed traffic volume is given for each Origin-Destination pair and equally assigned to the route candidates generated by the
Q Value-based Dynamic Programming with Boltzmann Distribution. The traveling
time of each section in the road network is evaluated by the well known Bureau of
Public Roads (BPR) volume-delay function in each iteration for converging the traffic
volume of the routes.

The comparison between the proposed method and the shortest path algorithms in
the simulation indicates that the proposed method could reduce the risk of the occur-
rence of the traffic congestion and save the traveling cost effectively. In addition, the
computation time is given to reveal the feasibility of the proposed method in large scale
networks.

This chapter is organized as follows: In the next section, the outline of Bureau of
Public Roads (BPR) volume-delay function is reviewed, while Q value-based Dynamic
Programming with Boltzmann distribution is described in section 3.3. The details of
the proposed procedure for analysis are described in section 3.4. In addition, section
3.5 shows the simulations and section 3.6 is devoted to conclusions.

### 3.2 BPR Volume-delay Function

In order to obtain the relationship between the traffic volume and traveling time,
the following most widely used volume delay function, BPR function [19][20], is used
to calculate the traveling time of each section from its traffic volume in this chapter.

\[
\begin{align*}
t_{ij} & \leftarrow t_{ij}^0 (1 + 0.15 \frac{v_{ij}}{c_{ij}}^4), \\
\end{align*}
\]

(3.1)

where,

- \(s_{ij}\): the section from intersection \(i\) to intersection \(j\)
- \(t_{ij}\): traveling time of section \(s_{ij}\)
- \(t_{ij}^0\): free flow traveling time of section \(s_{ij}\)
- \(c_{ij}\): traffic volume capacity of section \(s_{ij}\)
$v_{ij}$: traffic volume of section $s_{ij}$

### 3.3 Q value-based Dynamic Programming with Boltzmann Distribution

In the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution in chapter 2, Q values of the intersection pairs are updated using Eq.(2.1), in which the optimal route is selected by calculating $\arg\min_{i \in A(j)} Q_d(i, j)$, that is, the Greedy Strategy.

However, when Boltzmann Distribution is considered, which intersection should be selected as the next intersection is determined based on a certain probability. In this case, just using the minimum traveling time is not enough to represent the prospective traveling time to the destination.

Therefore, in the Q value-based Dynamic Programming with Boltzmann Distribution, the expectation of the traveling time for each Origin-Destination pair is calculated based on the following equations instead of the minimum traveling time in the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution.

Meanwhile, the definition of the Q value, i.e., $Q_d(i, j)$, also changes to the expectation of the traveling time to destination $d$, when a vehicle bound for destination $d$ moves to intersection $j$ at intersection $i$.

$$Q_d^{(n)}(i, j) \leftarrow t_{ij} + \sum_{k \in A(j)} P_{d}^{(n-1)}(j, k) Q_d^{(n-1)}(j, k),$$

(3.2)
where,

\( d \in D, \ i \in I - \{ d \} - B(d), \ j \in A(i) \)

\[
P_d^{(n)}(i, j) \leftarrow \frac{Q_d^{(n)}(i, j)}{\sum_{j \in A(i)} e^{-\frac{Q_d^{(n)}(i, j)}{\tau_d^{(n)}}}}, \tag{3.3}
\]

\( d \in D, \ i \in I - \{ d \} - B(d), \ j \in A(i) \)

\( Q_d^{(n)}(d, j) = 0, \quad d \in D, \ j \in A(d) \quad \tag{3.4} \)

\( Q_d^{(n)}(i, d) = t_{id}, \quad d \in D, \ i \in B(d) \quad \tag{3.5} \)

\( P_d^{(n)}(d, j) = 0, \quad d \in D, \ j \in A(d) - \{ d \} \quad \tag{3.6} \)

\( P_d^{(n)}(d, d) = 1.0, \quad d \in D \quad \tag{3.7} \)

\( i, j \in I : \) suffixes of intersections and their set

\( d \in D : \) suffix of destinations and its set

\( t_{ij} : \) traveling time from intersection \( i \) to intersection \( j \)

\( A(i) : \) set of suffixes of intersections moving directly from intersection \( i \)

\( B(i) : \) set of suffixes of intersections moving directly to intersection \( i \)

\( Q_d^{(n)}(i, j) : Q_d(i, j) \) in the \( nth \) iteration

\( P_d^{(n)}(i, j) : \) the probability that the vehicle bound for destination \( d \) moves to intersection \( j \) at intersection \( i \) in the \( nth \) iteration

\( \tau_d^{(n)}(d) : \) temperature parameter of intersection \( i \) when calculating the probability to the destination \( d \) in the \( nth \) iteration
3.4 Description of the proposed method

The flow chart of the traffic assignment strategy is shown in Fig. 3.1 and the detail of each module will be explained in the following subsections.

The basic idea of the traffic assignment strategy is to calculate the average traveling time over all the Origin-Destination pairs using Q Value-based Dynamic Programming with Boltzmann Distribution and compare it with the one using Q value-based Dynamic Programming, i.e., the greedy optimal route assignment method under the condition that the traffic system is static, where the fixed traffic volume is given for
each Origin-Destination pair.

Therefore, in addition to Q iterations another outer iteration is needed for obtaining the converged traffic volume and traveling time of each section as shown in Fig. 3.1 using the volume delay function.

Furthermore, a number of the route candidates from each Origin-Destination pair is generated using the probability for each section to be selected in order to evaluate the average traveling time over all Origin-Destination pairs. And also the special consideration for determining temperature $\tau_i^{(n)}(d)$ is studied in order to distribute the vehicles in the road network effectively.
3.4.1 Temperature Parameter

In this subsection, it is described how the temperature parameter of the Boltzmann Distribution in Eq.\((3.3)\) influences the evaluation of the proposed method. Let us take the road network in Fig.\(3.2\) for example. Fig.\(3.3\) shows the optimal traveling time and the optimal routes from each intersection to destination \(d\) calculated by the shortest path algorithm such as Dijkstra algorithm, A* algorithm, and Q value-based Dynamic Programming, while Fig.\(3.4\) displays the Q value and probabilities for each section to be selected, which are calculated by using Q value-based Dynamic Programming with Boltzmann Distribution under different temperature parameters. It is obvious that when the temperature \(\tau_i^{(n)}(d)\) is a small value, for example \(\tau_i^{(n)}(d) = 2\) in Fig.\(3.4(a)\), Q values are close to the optimal traveling time calculated by shortest path algorithms, and the route selection strategy is similar to the greedy strategy. Meanwhile, if the temperature \(\tau_i^{(n)}(d)\) is quite high, each section in the road network may have equal opportunities to be selected like Fig.\(3.4(c)\). Therefore, how to determine the temperature parameter plays a very important role in the proposed method.

Generally speaking, the temperature \(\tau_i^{(n)}(d)\) should be set according to the volume
Figure 3.3: The optimal routes from each intersection to the destination of the traffic system. Larger temperature should be adopted to distribute the traffic volume when traffic is heavy. However, the numerical range of Q values also influences the selection of $\tau^{(n)}(d)$. Even if the same constant $\tau^{(n)}(d)$ is used, the sections with a very small numerical range of Q values will be selected fairly randomly, conversely the ones with a large numerical ranges of Q values will tend to have the greedy strategy. Indeed, these kind of phenomena should be considered more carefully in the large scale road net work, where the numerical range of Q values of the sections are quite different with each other.

Thus, in order to distribute the vehicles without being influenced by the dynamic range of Q values in the whole road network, the temperature parameter in the proposed method is not a simple constant, but calculated based on the following average Q value over sections.

$$
\tau^{(n)}(d) = \tau_0 \left( \frac{\sum_{j \in A(i)} Q^{(n)}(i,j)}{|A(i)|} \right)^\theta,
$$

where,

$\tau_0, \theta : \text{constant}$
Figure 3.4: The Q values and probabilities of each section when using different temperature parameter $\tau_i^{(n)}$
When $\theta$ equals 0, the temperature parameter of all the intersections in the road network is equal to $\tau_0$, that is, constant. In this case, the intersections far from the destination, where the numerical ranges of $Q$ values are large, will take the greedier strategy than the intersections close to the destination, where the small numerical ranges of $Q$ values are used. Meanwhile, if $\theta$ equals 1, the situation will be contrary. How to select the value of $\theta$ and $\tau_0$ is carried out in the simulations of this chapter.

It seems from Fig.3.3 and Fig.3.4 that $Q$ value-based Dynamic Programming with Boltzmann Distribution is not better than the greedy strategy. It is because the traveling time of sections are fixed, in other words, the traffic volumes of sections are fixed. In this paper, it is studied how the proposed method outperforms the greedy method by changing the traveling times and traffic volumes of sections depending on the optimal route calculations.

### 3.4.2 Generate Route Candidates

In the proposed method, a considerable number of route candidates are generated based on the probabilities calculated by $Q$ value-based Dynamic Programming with Boltzmann Distribution for each OD pair. Then, the traffic volume will be equally assigned to these candidates. But, the candidate routes might be the same if necessary. As a result, the routes with small $Q$ values and large probabilities may be selected more than once and consequently attract more traffic volumes.

When the number of candidate routes is large enough, the probability of each section calculated by the candidate routes will be equal to the probability calculated by the proposed method. However, it should not be too large, since too many candidate routes may waste memory spaces and increase the computing time. Meanwhile, too small number of candidate routes are also not encouraged, since it’s necessary to introduce enough alternative routes to distribute the vehicles in the road networks. The
reasonable number of candidate routes is discussed in the simulations.

In addition, the cycles in the routes are forbidden.

### 3.4.3 Traffic Assignment

The traffic volumes are assigned to the candidate routes generated by the proposed method using the following equations.

\[
v_{ij}^{od} (w) \leftarrow \begin{cases} 
  v^{od} & \text{if } s_{ij} \in R_{od}(w) \\
  0 & \text{if } s_{ij} \notin R_{od}(w) 
\end{cases}, 
\]  

(3.15)

\[
v_{ij} \leftarrow \frac{1}{|W|} \sum_{w \in W} v_{ij}^{od}(w), 
\]

(3.16)

where,

- \( s_{ij} \): section from intersection \( i \) to intersection \( j \)
- \( v_{ij} \): traffic volume of section \( s_{ij} \)
- \( od \in OD \): suffix of OD pairs and its set
- \( w \in W \): suffix of candidate routes and its set
- \( v^{od} \): traffic volume of \( od \in OD \)
- \( v_{ij}^{od}(w) \): traffic volume of section \( s_{ij} \) when the vehicle of \( od \in OD \) takes the \( w \)th candidate route
- \( R_{od}(w) \): the \( w \)th candidate route of \( od \in OD \)
3.5 Simulation

3.5.1 Simulation Environment

In this paper, the proposed method are simulated using the Kitakyushu City’s map as shown in Fig.3.5, which includes 28,973 intersections and 42,284 bidirectional sections with four kinds of road speed limits, i.e., 30km/h, 50km/h, 60km/h and 80km/h. 20 origins and 20 destinations (400 OD pairs), which are mainly located in the center of the city, are also displayed in Fig.3.5. Static traffic volume is given to each OD pair.
Table 3.1: Average and deviation of $E$ with different combinations of $\theta$ and $\tau_0$

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$\tau_0$</th>
<th>Average of $E$ (sec)</th>
<th>Deviation of $E$ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>2405.74</td>
<td>66.75</td>
</tr>
<tr>
<td>0.1</td>
<td>5</td>
<td>2274.55</td>
<td>85.35</td>
</tr>
<tr>
<td>0.2</td>
<td>3</td>
<td>2220.31</td>
<td>44.79</td>
</tr>
<tr>
<td>0.3</td>
<td>2</td>
<td>2214.69</td>
<td>147.17</td>
</tr>
</tbody>
</table>

The average traveling time from origins to destinations over all the candidate routes, which is calculated by Eq.(3.17), is used to evaluate the two methods. In the conventional shortest path algorithm, the number of $|\mathcal{W}|$ will be considered as 1, since only one route is selected for one OD pair. The traveling time of each section is updated by BPR function in Eq.(3.1), where, the traffic volume capacity of each section is set at 3600 cars/h. What’s more, Q value-based Dynamic Programming is adopted as one of the shortest path algorithms to compare with the proposed method.

$$E \leftarrow \frac{1}{|\mathcal{OD}| |\mathcal{W}|} \sum_{od \in \mathcal{OD}} \sum_{we \in \mathcal{W}} \sum_{s_{ij} \in R_{od}(w)} t_{ij}$$  \hspace{1cm} (3.17)

### 3.5.2 Analysis on $\theta$ and $\tau_0$

In the proposed method, the temperature parameter plays a very important role. The value of $\tau_i^{(n)}(d)$ should be determined according to the traffic volume of the traffic system and the numerical range of the Q values.

As shown in Eq.(3.14), the parameter $\theta$ and $\tau_0$ are considered to determine $\tau_i^{(n)}(d)$. In this subsection, four different $\theta$ are simulated with changing $\tau_0$ as shown in Fig.3.6. The traffic volume of each OD pair is equal to 120 cars/h in the simulations. It seems that smaller $\theta$ with larger $\tau_0$ could perform as well as larger $\theta$ with smaller $\tau_0$. However, adopting too small $\theta$ with large $\tau_0$ or too large $\theta$ with small $\tau_0$ may make the intersections with small numerical ranges of the Q values lose their guidance, and con-
Figure 3.6: Analysis of $\theta$ with different $\tau_0$

Figure 3.7: Analysis of $\tau_0$ with different traffic conditions
Figure 3.8: Comparison of the average traveling time using different number of candidate routes

sequently cause unexpected random selections on these intersections. Therefore, the deviations of \( E \) by using the combinations of \( \theta \) and \( \tau_0 \) which performed best in Fig. 3.6 are calculated in Table 3.1. It shows the reasonable combinations of \( \theta \) and \( \tau_0 \) not only minimize the average traveling time, but also reduce the deviation, which means that the traffic system is more stable. Based on Table 3.1, \( \theta = 0.2 \) is selected in other simulations.

In addition, in order to reveal the relationship between the traffic volume and temperature parameter selection, the proposed method is simulated in three different traffic conditions, where the traffic volume of each OD pair is equal to 30 cars/h, 60 cars/h and 120 cars/h, respectively, changing \( \tau_0 \) as shown in Fig. 3.7. \( \theta \) is set at 0.2 and the number of candidate routes |\( W \)| is set at 100 during the simulation.

It is obvious from Fig. 3.7 that the best \( \tau_0 \) is small when the traffic volume of each OD pair is not large, since the distribution of the vehicles is not necessary, while \( \tau_0 \) becomes larger as the traffic is heavier. Therefore, introducing different \( \tau_0 \) may improve the efficiency of the proposed method when dealing with various traffic situations.
3.5.3 Analysis on the Number of Candidate Routes

As we known, introducing unnecessarily large number of candidate routes may sacrifice memory spaces and increase the computing time, while too small number of candidate routes may not include enough alternative routes to distribute the vehicles. Therefore, simulations with gradually changing $|W|$ are carried out in Fig.3.8 in order to find an appropriate number of candidate routes. The traffic volume of each OD pair and the temperature parameter are set at 120 cars/h and at $\tau_0 = 3$, respectively. The average of traveling time $E$ is used for the traffic volume and traveling time convergence.

The result indicates that the average traveling time of $E$ becomes smaller and finally converges as the number of candidate routes $|W|$ increases. Therefore, $|W| = 100$, which is large enough to guarantee the performance, is considered as a proper value for common use in the traffic networks.

3.5.4 Comparison of the Proposed Method and the Shortest Path Algorithm

In order to evaluate the efficiency of the proposed method when it is implemented in the large scale road network, the comparison between the proposed method and shortest path algorithm is carried out under three traffic conditions. The parameters used for each traffic condition is shown in Table 3.2.

As we can see from Fig.3.9, the performance of two methods is similar to each other when the traffic is not heavy, while the differences becomes larger as the traffic volume of each OD pair in the road network becomes larger. The results of the shortest path algorithm is changing alternately between two values, which indicates that the traffic volume is assigned to the shortest path and consequently the traffic jam is caused, and then turns to former routes after the update of the traveling time in
Table 3.2: Parameter setting under different traffic conditions

<table>
<thead>
<tr>
<th>Traffic volume for each OD pairs</th>
<th>30 cars/h</th>
<th>60 cars/h</th>
<th>120 cars/h</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_0 )</td>
<td>0.1</td>
<td>1.0</td>
<td>3.0</td>
</tr>
<tr>
<td>(</td>
<td>W</td>
<td>)</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3.3: Five congestion risk levels

<table>
<thead>
<tr>
<th>Risk level</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_{ij}/c_{ij} )</td>
<td>(0.1, 0.5]</td>
<td>(0.5, 1.0]</td>
<td>(1.0, 3.0]</td>
<td>(3.0, 5.0]</td>
<td>more than 5.0</td>
</tr>
</tbody>
</table>

In order to show the traffic distribution more clearly, we divided the congestion risk of the sections into five levels depending on the proportion of \( v_{ij} \) and \( c_{ij} \) as shown in Table 3.3. The average numbers of the sections in each level are calculated and shown in Fig.3.10. Sections where \( v_{ij}/c_{ij} \) is smaller than 0.1 are considered as very small traffic volumes, and the numbers of these sections are counted in Table 3.4, since they are too large to be displayed in Fig.3.10.

It could be seen from Fig.3.10 that although the differences between two methods are not the same in all the traffic conditions, the number of sections in low congestion risk levels of the proposed method is always larger than the shortest path algorithm, while the number of the sections in high congestion risk levels of the proposed method is much smaller than the shortest path algorithm. It is also shown from Fig.3.10(c) and Table 3.4 that some sections are involved into level 5 when using the shortest path algorithm, while the number of almost unused sections of the shortest path algorithm is much more than the proposed method. Actually, \( v_{ij}/c_{ij} \) of the section in the worst
The traffic volume of each OD pair is 30 cars/h

The traffic volume of each OD pair is 60 cars/h

The traffic volume of each OD pair is 120 cars/h

Figure 3.9: Comparison between the the proposed method and shortest path algorithm under different traffic conditions
(a) The traffic volume of each OD pair is 30 cars/h

(b) The traffic volume of each OD pair is 60 cars/h

(c) The traffic volume of each OD pair is 120 cars/h

Figure 3.10: The average number of the sections in five congestion risk levels
Table 3.4: The number of sections with very small traffic volumes

<table>
<thead>
<tr>
<th>Traffic volume for each OD pairs</th>
<th>30 cars/h</th>
<th>60 cars/h</th>
<th>120 cars/h</th>
</tr>
</thead>
<tbody>
<tr>
<td>The proposed method</td>
<td>83912</td>
<td>83539</td>
<td>80783</td>
</tr>
<tr>
<td>The shortest path algorithm</td>
<td>83916</td>
<td>83766</td>
<td>83635</td>
</tr>
</tbody>
</table>

Table 3.5: The size of the maps

<table>
<thead>
<tr>
<th>Number of intersections</th>
<th>map1</th>
<th>map2</th>
<th>map3</th>
<th>map4</th>
<th>map5</th>
<th>map6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16175</td>
<td>49109</td>
<td>97975</td>
<td>116920</td>
<td>153011</td>
<td>253077</td>
</tr>
<tr>
<td>Number of sections</td>
<td>48458</td>
<td>121024</td>
<td>215116</td>
<td>266830</td>
<td>374636</td>
<td>608028</td>
</tr>
</tbody>
</table>

condition when using the shortest path algorithm is larger than 8, which means that the traveling time of the section is 615 times longer than that of the free flow traveling time. Contrarily, only a few number of sections are involved into level 3 and level 4 when Q value-based Dynamic Programming with Boltzmann Distribution is adopted.

3.5.5 Computation time of the proposed method

In order to reveal the feasibility of the proposed method in large scale networks, the average computation time of Q value calculation by using Q value-based Dynamic Programming with Boltzmann Distribution is carried out in this subsection. The proposed method has been simulated on six different sizes of maps as shown in Table 3.5 by using Xeon E5310(1.60GHZ) with 4GB RAM. Since the temperature parameters also affect the calculation speed, the computation time of different maps with changing $\tau_0$ is described in Table 3.6. It is the average computation time for the convergence of Q values, which means the time of one iteration in Fig.3.9. $\theta$ is set at 0.2 and $G$ is equal to 100 during the simulations. The calculation time of traditional shortest path Dijkstra
Table 3.6: The computation time (ms) of different maps with changing $\tau_0$

<table>
<thead>
<tr>
<th>$\tau_0$</th>
<th>map1</th>
<th>map2</th>
<th>map3</th>
<th>map4</th>
<th>map5</th>
<th>map6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>730</td>
<td>546</td>
<td>1009</td>
<td>1174</td>
<td>1938</td>
<td>3355</td>
</tr>
<tr>
<td>2</td>
<td>1053</td>
<td>819</td>
<td>1527</td>
<td>1901</td>
<td>2899</td>
<td>3841</td>
</tr>
<tr>
<td>3</td>
<td>1377</td>
<td>1221</td>
<td>2117</td>
<td>2478</td>
<td>3821</td>
<td>4571</td>
</tr>
<tr>
<td>4</td>
<td>1429</td>
<td>1463</td>
<td>2472</td>
<td>3109</td>
<td>4711</td>
<td>5300</td>
</tr>
<tr>
<td>5</td>
<td>1336</td>
<td>1797</td>
<td>2935</td>
<td>3751</td>
<td>5487</td>
<td>5933</td>
</tr>
<tr>
<td>6</td>
<td>1533</td>
<td>1960</td>
<td>3358</td>
<td>4224</td>
<td>6072</td>
<td>7039</td>
</tr>
<tr>
<td>7</td>
<td>1615</td>
<td>2226</td>
<td>3525</td>
<td>4609</td>
<td>6712</td>
<td>7819</td>
</tr>
<tr>
<td>8</td>
<td>1618</td>
<td>2210</td>
<td>3671</td>
<td>4915</td>
<td>6700</td>
<td>8307</td>
</tr>
<tr>
<td>9</td>
<td>1609</td>
<td>2344</td>
<td>3894</td>
<td>5181</td>
<td>7219</td>
<td>9018</td>
</tr>
<tr>
<td>10</td>
<td>1770</td>
<td>2415</td>
<td>3952</td>
<td>5346</td>
<td>7373</td>
<td>9307</td>
</tr>
</tbody>
</table>

Table 3.7: The computation time (ms) of shortest path algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>map1</th>
<th>map2</th>
<th>map3</th>
<th>map4</th>
<th>map5</th>
<th>map6</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDP</td>
<td>106</td>
<td>160</td>
<td>301</td>
<td>363</td>
<td>582</td>
<td>1515</td>
</tr>
<tr>
<td>DA</td>
<td>1829</td>
<td>32219</td>
<td>207687</td>
<td>325281</td>
<td>577500</td>
<td>2582428</td>
</tr>
</tbody>
</table>

algorithm (DA) whose time complexity is $O(n^2)$ and the Q value-based dynamic programming (QDP) is shown in Table 3.7 for comparison. It is shown from Table 3.6 that the computation time of Q value-based Dynamic Programming with Boltzmann Distribution is acceptable even if the maps with very large sizes are used.

### 3.6 Conclusion

In this chapter, Q value-based Dynamic Programming with Boltzmann Distribution is systematically studied in the static system using real road networks – Kitakyushu City’s map. It is seen from simulations that the proposed method could reduce the congestion and save the traveling cost effectively comparing with the shortest path algorithms. In addition, the selection of the temperature parameter and the number of
the candidate routes are discussed. It seems that introducing appropriate temperature parameters for Q value-based Dynamic Programming with Boltzmann Distribution depending on the traffic conditions could improve the efficiency of the proposed method.
Chapter 4

Dynamic Traffic Management using Temperature Parameter Control in Q value-based Dynamic Programming with Boltzmann Distribution

4.1 Introduction

In this chapter, a dynamic traffic management model with two temperature parameter control strategies, i.e., Network Method and Intersection Method has been developed in order to apply the Q Value-based Dynamic Programming with Boltzmann Distribution to the dynamic traffic systems with time-varying traffic information and consequently reduce the traffic congestion in global perspective.

Network Method uses the same temperature parameter for the whole road network according to the global traffic situations, while Intersection Method adopts a different temperature parameter for each intersection based on the traffic situations of sections connected to the intersection.
In addition, the proposed dynamic traffic management model has been evaluated in a small size microscopic simulator in this chapter. The simulation results revealed that it is important to use the temperature parameter control in dynamic traffic systems and the proposed traffic management method can improve the system performance comparing with the conventional greedy method.

This chapter is organized as follows: In the next section, the outline of Q value-based Dynamic Programming with Boltzmann distribution is reviewed, while two temperature parameter control strategies, i.e., Network Method and Intersection Method are presented in section 3. In addition, section 4 shows the simulations, in which the comparison among two temperature parameter control strategies and Greedy strategy is carried out under dynamically changing traffic situations and different traffic information update interval. The final section is devoted to conclusions.

### 4.2 Structure of the Dynamic Traffic Management Model

As can be seen from Fig.4.1, all the vehicles select their routes to the destination depending on the temperature parameter, Q values and probabilities calculated by the proposed algorithm and follow the principle of Q value-based Dynamic Programming with Boltzmann distribution, which is described in the last chapter.

In addition, all the temperature parameter, Q values and probabilities are calculated every time the traffic information is updated.

As we discussed in the last two chapters, the temperature parameter plays a very important role in the Q value-based Dynamic Programming with Boltzmann Distribution. Basically, the probability of the next section to be selected is inversely related to Q values. However, the parameter "temperature" also has its influence. When the "temperature" is very high, Boltzmann Distribution is identical to the random distribution in which each section has equal opportunities to be selected. On the other hand, when
the "temperature" approaches 0, only the shortest path is available just like Greedy Strategy.

In this sense, it is better to adopt a small temperature parameter when the traffic is light and increase the temperature parameter as the traffic becomes heavier. It is not necessary to distribute the traffic when the number of the vehicles doesn’t exceed the capacity of the sections, while the vehicles should be assigned to different routes to avoid the traffic congestion when the number is large.

According to the well-know Wardrop equilibrium theory [21], the ideal condition is that the vehicles are assigned to different routes, however, no matter which route the drivers select, the traveling times of the vehicles from the origin to destination are equal with each other and no vehicles will gain from changing routes. Based on the above concept, two temperature parameter control strategies are proposed in the following subsections, which are Network Method and Intersection Method.

Figure 4.1: Flow chart of the dynamic traffic management model
4.3 Network Method

Network Method is the simplest temperature parameter control strategy, in which the identical temperature parameter is used in all sections for calculating the probability of selecting the next intersection depending on the total number of vehicles in the traffic system in the following.

\[
P^{(n)}_{d}(i, j) \leftarrow \frac{e^{Q^{(n)}_{d}(i, j)}}{\sum_{j \in A(i)} e^{Q^{(n)}_{d}(i, j)}},
\]

\[
\tau = \frac{\tau_{\text{max}}(N)}{1 + e^{-\alpha(NV - \beta)}},
\]

where,

\( \tau_{\text{max}}(N), \alpha, \beta : \) constant,

\( NV: \) the number of vehicles in the whole traffic system.

The principle of Network Method is to adjust the temperature parameter in global perspective. As shown in Fig.4.2, the Network Method increases the temperature parameter to provide more optimal routes for drivers when the whole traffic system is crowded, while just the Greedy Strategy is adopted if the total number of vehicles is small.

4.4 Intersection Method

Intersection Method adopts different temperature parameter for each intersection \( i \) based on \( W_i \), i.e., the sum of the longest waiting time of the sections connected to
intersection $i$, according to the following equations.

$$P_d^{(i,j)}(i, j) \leftarrow \frac{e^{-\frac{d_d^{(i,j)}}{\tau_i}}}{\sum_{j \in A(i)} e^{-\frac{d_d^{(i,j)}}{\tau_i}}}, \quad (4.3)$$

$$\tau_i = \frac{\tau_{\max}(I)}{1 + e^{-\mu(W_i - \theta)}}, \quad (4.4)$$

$$W_i = \sum_{j \in B(i)} W_{ji}, \quad (4.5)$$

where,

- $\tau_{\max}(I), \mu, \theta$: constant
- $W_{ji}$: the longest waiting time of the vehicles in section $s_{ji}$ during the last traffic information updating interval

As shown in Fig.4.3, the principle of Intersection Method is similar to Network Method. The value of the temperature parameter increases as the traffic becomes heavy.
Figure 4.3: Intersection Method

ier in order to distribute the traffic volumes. However, Intersection Method only distributes the vehicles at the intersections with overcrowded traffics, while Greedy Strategy is used at low traffic intersections. It enables enhanced routing strategies in the road network since each intersection has its own temperature parameter.

4.5 Simulation

4.5.1 Traffic Simulator

In this chapter, a traffic simulator for evaluating the proposed method is described. As can be seen from Fig.4.1, the temperature parameter, Q values and probabilities are calculated for the vehicles every time the traffic information is updated, that is $t_{ij}$, $W_{ij}$ and $W_i$ are updated depending on the time-varying traffics. The traveling time $t_{ij}$, which is initialized according to the distance of section $s_{ij}$, is updated by calculating the sum of the initialized traveling time and $W_{ij}$. 
A 7 × 11 road network, where the length of the sections in the road network is set at the range of 9 to 25 as shown in Fig. 4.4, is used in the simulator. Each road in the network is bidirectional and each section has two driveways as shown in Fig. 4.5, in which the vehicles going to turn left choose the left one, and vehicles going to turn right and turn around prefer the right one. The signal control follows the regulation shown in Table 4.1, in which the time delay between the neighboring intersections is 3 time steps.

The occurrence of the vehicles in each section has the Poisson distribution with different occurrence rate $\lambda_{ij}$.
Figure 4.5: An example of intersections in the road network

\[
p_{ij}(n) = \frac{(\lambda_{ij}s)^n e^{-\lambda_{ij}s}}{n!}, \quad n = 0, 1, ... \quad (4.6)
\]

where,

\( p_{ij}(n) \): the probability that \( n \) vehicles occur in section \( s_{ij} \) during \( s \) time steps

\( \lambda_{ij} \): the rate that vehicles occur at section \( s_{ij} \) (the number of vehicle occurrence/unit time step).

All the vehicles in the simulator determine their routes to their destination based on the Q values and probabilities calculated by Q value-based Dynamic Programming with Boltzmann Distribution when they occurred at the origin. The speed limit for each section and the kind of vehicles are the same, but the vehicles could move one step forward only if there are no other vehicles in front of them. In addition, the loops in the routes are forbidden.
(a) Average traveling time under different traffic situations

(b) Average waiting time under different traffic situations

Figure 4.6: Comparison of different temperature parameters
4.5.2 Temperature Parameter Analysis of Network Method

In this subsection, six constant temperature parameters of Network Method are tested under three different traffic situations, i.e., Small, Middle and Large, which are defined in Table 4.2. For each traffic situation, we simulated twenty times using the same vehicle occurrence rate, but different random seeds. The average traveling time from the origin to destination and average waiting time, i.e., average stopping time over all the vehicles during the whole simulation time period are calculated to evaluate the system performance. Other simulation conditions are given in Table 4.3. Fig.4.6 shows the average traveling time and average waiting time with different temperature parameters.

As shown in Fig.4.6, the value of the best temperature parameter is small in the light traffic, in which the distribution of the vehicles is not necessary, while it becomes large as the traffic becomes heavier. Meanwhile, the comparison between Fig.4.6(a) and Fig.4.6(b) shows that the temperature parameter with unnecessarily large values increases the traveling time and waiting time a lot since it forces the drivers to drive randomly in the traffic system. Considering the above, it is clear that adjusting the temperature parameter depending on the traffic situations is necessary and we decided to increase the temperature parameter from one to ten as the number of the vehicles NV

Table 4.2: Traffic situations in simulations

<table>
<thead>
<tr>
<th>Traffic situation</th>
<th>Occurrence rate in external sections (vehicles/time step)</th>
<th>Occurrence rate in internal sections (vehicles/time step)</th>
<th>Average of $N$ (vehicle)</th>
<th>Average of $W_{ij}$ (time step)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>0.02</td>
<td>0.005</td>
<td>200-500</td>
<td>1-2</td>
</tr>
<tr>
<td>Middle</td>
<td>0.08</td>
<td>0.005</td>
<td>500-1000</td>
<td>2-4</td>
</tr>
<tr>
<td>Large</td>
<td>0.13</td>
<td>0.005</td>
<td>1000-1500</td>
<td>4-5</td>
</tr>
<tr>
<td>Very Large</td>
<td>0.18</td>
<td>0.005</td>
<td>more than 1500</td>
<td>more than 5</td>
</tr>
</tbody>
</table>
Table 4.3: Parameter setting for simulations

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum time step</td>
<td>3000</td>
</tr>
<tr>
<td>Information update interval step</td>
<td>10</td>
</tr>
<tr>
<td>Initial number of vehicles</td>
<td>200-250</td>
</tr>
</tbody>
</table>

Table 4.4: Parameter setting for Network Method

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{\text{max}}(N)$ coefficient of Network Method</td>
<td>10</td>
</tr>
<tr>
<td>$\alpha$ coefficient of Network Method</td>
<td>0.0035</td>
</tr>
<tr>
<td>$\beta$ coefficient of Network Method</td>
<td>600</td>
</tr>
</tbody>
</table>

changes from 200 to 1500 in Network Method. Therefore, the parameters of Network Method are set as shown in Table 4.4.

4.5.3 Temperature Parameter Analysis of Intersection Method

In order to determine how to adjust the temperature parameters of Intersection Method, four different sets of parameters of Intersection Method shown in Fig.4.7(a) are simulated under three different traffic situations in this subsection. For each parameter set, we also simulated twenty times with different random seeds and the average traveling time and waiting time of all the vehicles are calculated to do the evaluation. Fig.4.7(b) and Fig.4.7(c) show that set 2, in which $\tau_{\text{max}}(I) = 10$, $\mu = 0.040$ and $\theta = 40$, is the best selection for the parameters in Intersection Method.

4.5.4 Comparison of Various Temperature Control Strategies

The comparison among Network Method, Intersection Method and Greedy Method is carried out in this subsection.
Different parameter settings in Intersection Method

Average traveling time under different traffic situations

Average waiting time under different traffic situations

Figure 4.7: Comparison of different parameter setting in Intersection Method
Firstly, three methods with various traffic information updating intervals is simulated under Small, Middle and Large traffic situations. Each Method is simulated twenty times with different random seeds and the average traveling time and waiting time of all the vehicles shown in Fig.4.8, Fig.4.9 and Fig.4.10 are calculated to evaluate the efficiency of the three methods. As we can seen from the results, the average traveling time and waiting time increase as the traffic information updating interval becomes longer, but the degrees of the increase are different. When the ideal traffic information updating interval is adopted, i.e., the update interval is very short, even Greedy Method could perform well, while the proposed methods, especially the Network Method, presents their outstanding performance when the traffic information couldn’t be updated so often.

Another simulation is carried out for 10000 time steps with dynamically changing traffic situations shown in Fig.4.11(a). Fig.4.11(b) shows how the total waiting of all the sections changes as the time step goes when adopting different routing strategies with the information update interval of 10 time steps. In addition, the average traveling time of all the vehicles during the simulation by Greedy Method, Network Method and Intersection Method are equals to 251.1 time steps, 230.8 time steps and 218.7 time steps, respectively. These results indicate that Intersection Method performs the best when dealing with the heavy traffic situations.

### 4.6 Conclusion

In this chapter, a dynamic traffic management strategy using the temperature parameter control in the Q value-based Dynamic Programming with Boltzmann Distribution has been proposed. Two temperature parameter control strategies, i.e., Network Method and Intersection Method are studied and compared with the conventional Greedy Method in the simulations. As can be seen from the simulation results, it is
Figure 4.8: Comparison of three methods under different information update interval in Small traffic situation.
Figure 4.9: Comparison of three methods under different information update interval in Middle traffic situation
Figure 4.10: Comparison of three methods under different information update interval in High traffic situation

(a) Average traveling time under different information update interval

(b) Average waiting time under different information update interval
(a) Vehicle occurrence rate $\lambda_{ij}$

(b) Sum of the longest waiting time $W_{ij}$ of all the sections

Figure 4.11: Comparison of different strategies
clarified that the proposed methods are simple, but useful for improving the efficiency of the traffic system and reducing the traffic congestion comparing with the Greedy Method.

As for the application of the proposed method to real traffic systems, the proposed method including the temperature parameter control strategies should be extended in terms of the applicability to large scale real road networks.
Chapter 5

An application of Q value-based Dynamic Programming with Boltzmann Distribution to real world road networks

5.1 Introduction

In this chapter, a further advanced dynamic traffic management model is finally proposed and applied to the large scale microscopic simulator SOUND/4U based on the real world road network of Kurosaki, Kitakyushu in Japan. All the vehicles in the simulator follow the direction from the route guidance of the dynamic traffic management model, in which the extended Q Value-based Dynamic Programming with Boltzmann Distribution and the time-varying traffic information are used to generate the routes from the origins to destinations. The simulation results show that the proposed Q Value-based Dynamic Programming with Boltzmann Distribution could reduce the traffic congestion and improve the efficiency of the whole traffic system effectively.
Figure 5.1: Structure of dynamic traffic management model

This chapter is organized as follows: In the next section, the detail of the proposed dynamic traffic management model is explained, while section 5.3 shows the simulations, in which the comparison between the proposed method and greedy method is carried out under dynamically changing traffic situations. Section 5.4 is devoted to conclusions.

5.2 Dynamic Traffic Management Model

As shown in Fig. 5.1, there are three modules, i.e., real-time traffic data collector, route guidance module and system evaluation module in the proposed dynamic traffic management model. Each module will be explained in detail in the following subsections.
5.2.1 Real-time traffic data collector

In this chapter, a large scale microscopic simulator SOUND/4U is used to replicate the complex traffic flow dynamics and implement different traffic management strategies in the urban-level real world road network. The real-time traffic data collector is also implemented based the SOUND/4U simulator. For every minute, the traffic data collector gets the traffic information from the vehicles in the traffic simulation system, which includes vehicles’ starting time from the origin, arrival time to the destination and the time entered and left each section, and send them to the route guidance module and system evaluation module.

5.2.2 Route guidance module

As shown in Fig. 5.1, all the vehicles in the traffic systems follow the directions from the route guidance module. The route guidance module firstly constructs the logistic network based on the geographical road network and then calculates the traveling cost for each section in the logistic network based on the traffic information from the traffic data collector. In addition, routes from the origin to destination of vehicles are generated using the Q values and the probabilities calculated by Q value-based Dynamic Programming with Boltzmann Distribution.

<Logistic network constructing>

In order to calculate the optimal routes for the vehicles using a graph search algorithm, the geographical road network should be transformed into the logistic graph. The simplest way is to consider the crosses in the road network as the intersections and the roads as the directed sections. However, in the real world traffic system, the vehi-
Figure 5.2: Example of logistic network construction

(a) cross in geographical road network

(b) cross in logistic network
cles decided to turn left, turn right or go straight may spend different traveling time to go through certain roads even if they are the same road, due to the signal, traffic rules and the number of the vehicles in that direction. Therefore, the vehicles going through different directions are considered as traveling on different sections in the logistic network in this paper. Fig. 5.2 shows an example on how general three directions in the intersections are transformed into the logistic network. The traveling time $t_{ij}$ from intersection $i$ to intersection $j$ in the logistic network is dynamically changing depending on the traffic information sent from the traffic data collector according to the following equation.

$$t_{ij} = \begin{cases} 
\frac{\sum_{v \in \mathcal{V}(ij)} (lt_{ij}(v) - et_{ij}(v))}{|\mathcal{V}(ij)|}, & \text{if } |\mathcal{V}(ij)| \neq 0, \\
\frac{\sum_{v \in \mathcal{V}(ij)} t'_{ij}(v)}{|\mathcal{C}(ij)|}, & \text{if } |\mathcal{V}(ij)| = 0, \\
t^0_{ij}, & \text{if } |\mathcal{C}(ij)| = 0,
\end{cases} \tag{5.1}$$

where,

$t'_{ij} = ct - et_{ij}(v), \quad \text{if } (ct - et_{ij}(v)) > t^0_{ij}$
$t'_{ij} = t^0_{ij}, \quad \text{if } (ct - et_{ij}(v)) \leq t^0_{ij}$

$lt_{ij}(v)$: time when vehicle $v$ left section $s_{ij}$
$et_{ij}(v)$: time when vehicle $v$ entered section $s_{ij}$
$\mathcal{V}(ij)$: set of suffixes of vehicles going through $s_{ij}$ during the last traffic information updating interval
$\mathcal{C}(ij)$: set of suffixes of vehicles in section $s_{ij}$ at the current traffic information updating interval
$ct$: the current time
Figure 5.3: A simple network

$t_{ij}^0$: free flow traveling time of section $s_{ij}$

**<Q values and the probabilities calculation>**

Q value-based Dynamic Programming with Boltzmann Distribution described in the chapter 3 is used to calculate the Q value, i.e., the expected traveling time from each origin to the destination and the probability of the next section to be selected in the route guidance module.

However, the following equations, in which $EQ_d^{(n)}(i)$ is added, is used to calculate the $P_d^{(n)}(i, j)$ instead of the conventional one in Eq. (3.3).

$$P_d^{(n)}(i, j) \leftarrow \frac{e^{-EQ_d^{(n)}(i)}}{\sum_{j \in A(i)} e^{-EQ_d^{(n)}(j)}}$$

(5.2)
The reasons why this modification is necessary is explained as follows.

As we discussed in the chapter 3, in the large scale road network, the numerical range of Q values are quite different with each other. Generally speaking, the Q val-
ues of the intersection pairs near the destinations are much smaller than the Q value of the intersection pairs far away from the destinations. In the case that constant $\tau$ like Eq. (5.2) is adopted, the conventional Q value-based Dynamic Programming with Boltzmann Distribution may introduce the unexpected greedier strategy for the intersection pairs with large Q values comparing with the intersection pairs with small Q values no matter what's the value of $\tau$.

Let us take the simple extreme road network in Fig. 5.3 for example, where the
numbers show the traveling time of sections. Fig. 5.4 and Fig. 5.5 show the Q values and the probabilities calculated by the conventional method and the new proposed method with different $\tau$, respectively. As shown in Fig. 5.4, although $Q_{d}(f, e)$ is twice of $Q_{d}(f, d)$, $s_{fe}$ still has the probability of 23% to be selected, while only 47% for $s_{fd}$. Contrarily, the probability of selecting $s_{ob}$ is 28% and that of selecting $s_{oc}$ is 66% despite $Q_{d}(o, b)$ is almost the same as $Q_{d}(o, c)$.

Actually, in the dynamic traffic system, it is not encouraged to select the routes randomly for the intersection pairs with small Q values. Intersection pairs with small numerical ranges of Q values mean not only that their locations are close to the destination, but also that the traffic volume on the routes to the destination is not heavy. Therefore, the strategy like the greedier ones should be adopted because it is not necessary to distribute the traffic volume when the traffic does not exceed route capabilities.

Considering the above reasons, $EQ_{d}(i)$ is introduced into Q value-based Dynamic Programming with Boltzmann Distribution in this chapter, which solve the above problems as shown in Fig. 5.5.

How to determine a suitable parameter $\tau$ is discussed in the simulation of this chapter.

<Routes generation>

Based on the probabilities calculated by Q value-based Dynamic Programming with Boltzmann Distribution, the routes from the origin to destination are generated for each vehicle in the traffic system. Let’s use the results in Fig. 5.5(a) to explain how to generate the routes. Suppose there is a driver who wants to drive from origin $o$ to destination $d$, then the probability of this driver to visit section $s_{oa}$ is 8% and 31% for $s_{ob}$ and 61% for $s_{oc}$. If there are 100 vehicles trying to travel from $o$ to $d$, there will be
8 vehicles going through $s_{oa}$, 31 vehicles going through $s_{ob}$ and 61 vehicles traveling from $o$ to $c$ in the ideal case.

In addition, the loops in the generated routes will be forbidden.

### 5.2.3 System evaluation module

Three kinds of data are available to evaluate the system performance from the traffic data collector.

First one is the traveling time of the vehicles from the origin to destination, which is calculated as follow.

$$
    t_v = \begin{cases} 
    dt(v) - ot(v), & \text{if vehicle } v \text{ arrived the destination}, \\
    ft - ot(v), & \text{if vehicle } v \text{ didn't arrived the destination} \quad \text{when simulation ended}, 
    \end{cases} 
$$

where,

$t_v$: total traveling time of vehicle $v$ during the simulation

$dt(v)$: time when vehicle $v$ arrived the destination

$ot(v)$: time when vehicle $v$ started from the origin

$ft$: the time when simulation ended

Second one is the number of the vehicles $nv_{ij}$ congested in section $s_{ij}$ in the road network.

$$
    nv_{ij} = \sum_{v \in V(ij) \cup CV(ij)} J(v), 
$$
Table 5.1: Data of road network

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of intersections</td>
<td>4243</td>
</tr>
<tr>
<td>Number of sections</td>
<td>7941</td>
</tr>
<tr>
<td>Number of OD</td>
<td>20</td>
</tr>
<tr>
<td>Number of OD pairs</td>
<td>380</td>
</tr>
</tbody>
</table>

\[
J(v) = \begin{cases} 
1, & \text{if } (lt_{ij}(v) - et_{ij}(v)) > t^0_{ij} \text{ for } v \in V(ij) \\
& \text{or } (ct - et_{ij}(v)) > t^0_{ij} \text{ for } v \in CV(ij), \\
0, & \text{if } (lt_{ij}(v) - et_{ij}(v)) \leq t^0_{ij} \text{ for } v \in V(ij) \\
& \text{or } (ct - et_{ij}(v)) \leq t^0_{ij} \text{ for } v \in CV(ij),
\end{cases} \tag{5.6}
\]

The third one is the traveling time of the vehicles going through section \( s_{ij} \).

\[
t_{ij}(v) = lt_{ij}(v) - et_{ij}(v) \tag{5.7}
\]

5.3 Simulation

5.3.1 Simulation Condition

In this chapter, the Q Value-based Dynamic Programming with Boltzmann Distribution has been applied to the SOUND/4U simulator based on the road network of Kurosaki, Kitakyushu in Japan. The data of the road network is shown in Table 5.1. As shown in Fig.5.6, 20 Origins and 20 Destinations (OD) are picked up in the road net-
The vehicle occurrence rate of each OD pair is assumed to follow the following Poisson distribution.

\[ p_{od}(n) = \frac{(\lambda_{od}s)^n e^{-\lambda_{od}s}}{n!}, \quad n = 0, 1, ... \]  

(5.8)

where,

- \( p_{od}(n) \): the probability that \( n \) vehicles depart from origin \( o \) to destination \( d \) during time \( s \)
- \( \lambda_{od} \): the average rate that the vehicle departs from origin \( o \) to destination \( d \) in unit time (the number of vehicles/minute)
Figure 5.7: Comparison of temperature parameters in different traffic situations

(a) Average $\lambda_{od}$ of all the OD pairs = 0.2

(b) Average $\lambda_{od}$ of all the OD pairs = 0.5

(c) Average $\lambda_{od}$ of all the OD pairs = 1
5.3.2 Temperature Parameter Analysis

The temperature parameter $\tau$ in Eq.(5.2) is a parameter that could affect the relationship between the Q values and probabilities. Basically, the distribution tends to be random as the parameter $\tau$ becomes larger. Adopting too small $\tau$ is identical to selecting the routes greedily, while too large $\tau$ may force the vehicles to travel without directions in the road network. In order to distribute the traffic volume effectively, it is crucial to select reasonable temperature parameter $\tau$ for the Q value-based Dynamic Programming with Boltzmann Distribution.

In this paper, $\tau$ is determined by simulations on different traffic situations as shown in Fig.5.7. For each traffic situations, we simulated three times and the time length for each simulation is 2 hour. The real-time traffic data collector sends the time-varying traffic information to the route guidance module for every 10 minutes. The average traveling time $t_v$ from the origin to destination is used to evaluate the performance of each parameter setting. As is indicated in Fig.5.7, the figures have the minimum average traveling time at a certain $\tau$ when the traffic is low or just middle, while the optimal parameter $\tau$ is hard to determine when the traffic is very heavy. Meanwhile, too large $\tau$ may sacrifice the traveling time a lot in any traffic situation, so forcing the drivers to drive fairly randomly in the traffic system is not an encouraging strategy.

The temperature parameter with the best average performance, i.e., $\tau = 0.002$, is selected in the simulations of the next subsections. It means that the traffic distribution strategy follows the following rules.

- Select the route greedily in the intersections with small average Q values.

- Distribute the traffic volume to each section depending on the Q values of in the intersections with comparatively large average Q values.

- Distribute the traffic volume equally to the sections with similar average Q values at any intersection.
5.3.3 Comparison between the proposed method and the greedy method

Figure 5.8: Average $\lambda_{od}$ of different OD sets in Experiment 1

Table 5.2: OD sets

<table>
<thead>
<tr>
<th>Set</th>
<th>OD</th>
</tr>
</thead>
<tbody>
<tr>
<td>OD set1</td>
<td>No2, No4, No6, No13, No16;</td>
</tr>
<tr>
<td>OD set2</td>
<td>No1, No3, No18, No19, No20;</td>
</tr>
<tr>
<td>OD set3</td>
<td>No5, No7, No8, No9, No10, No11, No12, No14, No15, No17.</td>
</tr>
</tbody>
</table>

In this subsection, the comparison between the proposed method and greedy method is carried out under two different traffic conditions. As shown in Table 5.2, the 20 ODs have been divided into 3 different OD sets and $\lambda_{od}$ of each OD set is shown in Fig.5.8 and Fig.5.9 in two experiments, respectively. Each experiment lasts 5 hours and the real-time traffic data collector sends the time-varying traffic information to the route guidance module every 10 minutes. The average $t_i$ (Eq.5.4) of the vehicles arrived the
Figure 5.9: Average $\lambda_{od}$ of different OD sets in Experiment 2

Table 5.3: Comparison between the proposed method and the greedy method in two experiments

<table>
<thead>
<tr>
<th>Item</th>
<th>Experiment 1</th>
<th></th>
<th>Experiment 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Greedy</td>
<td>Proposed</td>
<td>Greedy</td>
<td>Proposed</td>
</tr>
<tr>
<td>Number of departed vehicles</td>
<td>59411</td>
<td>53302</td>
<td>51175</td>
<td>52124</td>
</tr>
<tr>
<td>Number of arrived vehicles</td>
<td>49691</td>
<td>53907</td>
<td>51175</td>
<td>52124</td>
</tr>
<tr>
<td>Average $t_v$ of all the vehicles</td>
<td>2428$s$</td>
<td>1638$s$</td>
<td>1749$s$</td>
<td>1349$s$</td>
</tr>
</tbody>
</table>
(a) Average $t_v$ of the vehicles arrived the destinations

(b) The total number of $n_v_{ij}$ of all the sections

(c) Average $t_{ij}(v)$ of the vehicles going through section $s_{ij}$

Figure 5.10: Comparisons in Experiment 1
(a) Average $t_v$ of the vehicles arrived the destinations

(b) The total number of $nv_{ij}$ of all the sections

(c) Average $t_{ij}(v)$ of the vehicles going through section $s_{ij}$

Figure 5.11: Comparisons in Experiment 2
destinations, the total number of $n_{ij}$ (Eq. (5.5)) over all the sections, and the average $t_{ij}(v)$ (Eq. (5.7)) of the vehicles going through the sections during the simulation time are used and shown to evaluate the two methods in Fig. 5.10 and Fig. 5.11. The number of departed vehicles from the origin, the number of arrived vehicles to the destination and the average $t_v$ of all the vehicles in the two experiments are shown in Table 5.3. The results indicate that the proposed Q value-based Dynamic Programming with Boltzmann Distribution could reduce the traffic congestion and improve the efficiency of the whole traffic system effectively compared with the greedy strategy in the real world road network.

5.4 Conclusion

In this chapter, a dynamic traffic management model has been proposed, which aims at alleviating the traffic congestion and improving the efficiency of the traffic systems in global perspective. The proposed traffic management model is applied to the large scale microscopic simulator SOUND/4U based on the road network of Kurosaki, Kitakyushu in Japan. All the vehicles in the simulator follow the direction from the route guidance module of the dynamic traffic management model, in which the extended Q value-based Dynamic Programming with Boltzmann Distribution and the time-varying traffic information are used to generate routes from the origins to destinations. The simulation results comparing the proposed method with the conventional greedy routing method shows that the proposed Q value-based Dynamic Programming with Boltzmann Distribution could reduce the traffic congestion and improve the efficiency of the whole traffic system effectively in the real world road network.
Chapter 6

Conclusions

In this thesis, a traffic management method, i.e., Q Value-based Dynamic Programming with Boltzmann Distribution has been proposed and systematically studied in order to find a good approximation to the global optimum of the traffic system.

The research is carried out by gradually studying the performance of the proposed method from simple static simulation to large scale real-time simulation.

Firstly, the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution, in which Q Value-based Dynamic Programming and Boltzmann distribution are simply combined to minimize the total traveling time of all the vehicles considering the traffic volumes, is proposed and applied to a small scale road network based on a static traffic assignment model and a dynamic traffic assignment model, respectively in chapter 2.

Secondly, the Q Value-based Dynamic Programming with Boltzmann Distribution, which is derived from the quasi-Q Value-based Dynamic Programming with Boltzmann Distribution, is described and applied to a large scale road network, i.e., the Kitakyushu City road network, based on the static traffic assignment model in chapter 3.

Thirdly, chapter 4 applied the Q Value-based Dynamic Programming with Boltzmann Distribution to the dynamic traffic systems with time-varying traffic information.
by adding two temperature parameter control strategies, i.e., Network Method and Intersection Method, and simulated them in a small size microscopic simulator.

Finally, a further advanced dynamic traffic management model is proposed and applied to the large scale microscopic simulator SOUND/4U based on the real world road network of Kurosaki, Kitakyushu in Japan in chapter 5.

The simulation results in these four chapters revealed the feasibility of the proposed method to be applied to the real world large scale road networks with time-varying traffic information and also proved that the proposed method performs well on reducing the risk of the occurrence of the traffic congestion, saving the traveling cost and is less affected by the frequency of the real time traffic information collection.
References


REFERENCES


Acknowledgements

Many people have offered me valuable help in my thesis writing. Firstly, I would like to give my sincere gratitude to Professor Hirasawa, my supervisor, for his extraordinary patience and consistent encouragement. He has walked me through all the stages of writing this thesis. Without his strong support, this thesis could not have reached its present form. He has offered me valuable ideas, suggestions and criticisms with his profound knowledge in forensic language and rich research experiences.

Secondly, my great respect and heartfelt thanks also go to Prof. Yoshie and Prof. Fujimura of Waseda University, who make efforts to provide me lots of helpful comments and advices as the co-examiners.

Thirdly, I would like to express my gratitude to Dr. Mabu and Dr. Shimada, whose patient attitude and kindness helped me a lot to the completion of my doctor courses and this thesis. Thanks are also due to my friends in our laboratory, who helped me to discuss the research and overcome the troubles. Special thanks should go to Ms. Yu Lu, Ms. Zhou Huiyu, Mr. Bing Li, Mr. Feng wen, Mr. Manoj Kanta Mainali, Mr. Qinbiao Meng, Mr. Deng Zhang and Mr. Fengming Ye for their encouraging me when I had problems carrying out the research and writing this thesis.

Last my thanks would go to my supervisor, Professor Songnian Yu in Shanghai university. Without his recommendation I wouldn’t have the chance to study in Waseda university. And also my beloved family for their loving considerations and great confidence in me all through these years.
Research Achievements

Journal Paper


International Conference Paper


2010/08.


