Game Theoretic Approach for the Design of Intelligent Traffic Networks

Final Report by
Srinivasan Ravichandran
under the guidance of Prof. Y. Narahari
at the Department of Computer Science and Automation
Indian Institute of Science, Bangalore

July 22, 2013
Acknowledgements

I would like to thank Prof. Narahari for taking me in as an intern and the Indian Academy of Sciences for providing me this excellent opportunity. This internship was a great learning experience with my team-mate Sacchit and our mentor Dilpreet Kaur. I would also like to use this opportunity to thank all the Professors and senior students who were helpful in the application process for this internship.
## Contents

1. **Basic Definitions and Models**
   - 1.1 Strategic Form Game ........................................ 6
   - 1.2 Model for Traffic Networks ................................. 6
   - 1.3 Strongly Dominant Strategy ................................. 7
   - 1.4 Nash Equilibrium ............................................ 7
   - 1.5 Correlated Strategy .......................................... 8
   - 1.6 Correlated Equilibrium ..................................... 8

2. **Equilibria in Transportation Networks** .......................... 9
   - 2.1 Nash Equilibrium Approach ................................. 10
     - 2.1.1 Computation .............................................. 10
     - 2.1.2 Performance ............................................. 10
     - 2.1.3 Programming ............................................ 11
   - 2.2 Correlated Equilibrium Approach .......................... 11
     - 2.2.1 Computation .............................................. 12
     - 2.2.2 Performance ............................................. 13
     - 2.2.3 Programming ............................................ 13

3. **Applications** .................................................. 14
   - 3.1 Alleviating network congestion ............................. 14
     - 3.1.1 The Internet .............................................. 14
     - 3.1.2 Real World Traffic Networks .......................... 14
   - 3.2 Braess’s Paradox .............................................. 15
     - 3.2.1 The Internet .............................................. 16
     - 3.2.2 Real World Traffic Networks .......................... 16
4 Conclusion

References
Abstract

Congestion in traffic networks has been on the rise and in the current state of affairs, a naive system does not meet the needs. An intelligent system is necessary to regulate traffic and to alleviate congestion. Such intelligent systems can be designed with the help of game-theoretic concepts such as Nash equilibria and correlated equilibria. The application of game-theoretic models for the analysis of such traffic networks and their efficient management is the main scope of this project. In this project, we attempt to determine a traffic distribution with the help of the equilibrium concept, such that all the users are benefited.
Introduction The study of intelligent transportation networks is one of the most pursued fields at the moment. This is because of the need to manage the traffic in the ever-growing Internet and also in real-world traffic networks. In this project, an attempt is made to obtain an equilibrium distribution of traffic in the network so that congestion in the network is alleviated.

1 Basic Definitions and Models

1.1 Strategic Form Game

A strategic form game $\Gamma$ is a tuple $\langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ where $N = \{1, 2, 3, ..., n\}$ is the finite set of players, $S_1, S_2, S_3, ..., S_n$ are the sets of strategies available to players $1, 2, 3, ... n$ respectively and $u_i : S_1 \times S_2 \times S_3 \times .... \times S_n \to \mathbb{R}$, for $i = 1, 2, 3, ... n$ are mappings that are called the utility or payoff functions.

1.2 Model for Traffic Networks

We represent transportation networks as directed graphs. Kleinberg\cite{2} has shown that a traffic network can be modeled as a strategic form game as defined above, where the players are the users of the traffic network, their strategies are the paths that they can use in the traffic network and the time of travel for each player can be considered as the payoff for each player. This model is being used here.

We model the traffic network as a game as follows. $N$ be the set of all players of the game(users of the network) and let $|N| = n$. Each player $p$ has a finite set of strategies $S_p$ with $|S_p| \geq 2$ and let the utility of the player $p$ be denoted as $u^p$.  

6
1.3 Strongly Dominant Strategy

A strategy \( s^*_i \in S_i \) is said to be a strongly dominant strategy for player \( i \) if it strongly dominates every other strategy \( s_i \in S_i \). That is, \( \forall s_i \neq s^*_i \),

\[
u_i(s^*_i, s_{-i}) > u_i(s_i, s_{-i}) \quad \forall s_{-i} \in S_{-i}\]

1.4 Nash Equilibrium

Before defining the concept of Nash equilibrium, it is necessary to understand the term strategy profile. A strategy profile is a collection of strategies, one for each player, where each player chooses his/her strategy from their strategy sets only. Thus, \( S = S_1 \times S_2 \times ....S_n \) is the set of all possible strategy profiles.

Now, a Nash equilibrium can be defined as follows.

**Definition** A Nash equilibrium in a strategic form game is a strategy profile \( s^* \in S \) and \( s^* = \langle a^*_1, a^*_2, ..., a^*_i, ..., a^*_j, ..., a^*_n \rangle \) such that no player \( i \) can do better by choosing an action different from \( a^*_i \), given that every other player \( j \) adheres to \( a^*_j \).

Nash equilibrium can be seen as that strategy profile where each user’s strategy is the best response against the best response of all the other players. Hence, no user will have the incentive to unilaterally deviate from his/her strategy.
1.5 Correlated Strategy

Before we go into the details of correlated strategies, we will first define a distribution.

A distribution $x$ on $S = S_1 \times S_2 \times \ldots \times S_n$ is a linear vector of non-negative real numbers and every element of $x$ corresponds to each strategy profile and the sum of the elements of $x$ is 1.

**Definition** A correlated strategy for a non-empty subset $C$ (also called coalition) of the players is any probability distribution over the set of possible combinations of pure strategies that these players can choose.

1.6 Correlated Equilibrium

A correlated equilibrium is any correlated strategy for the players which could be self-enforceingly implemented with the help of a mediator who can make non-binding recommendations to each player.

More formally, it can be defined as follows.

**Definition** Given a finite strategic form game, $\Gamma = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$, a correlated strategy $\alpha \in \Delta(S)$ recommended by a mediator is called a correlated equilibrium if

$$u_i(\alpha) = \sum_{s \in S} \alpha(s_i, s_{-i}) u_i(s_i, s_{-i}) \geq \sum_{s \in S} \alpha(s_i, s_{-i}) u_i(\delta_i(s_i), s_{-i}) \; \forall \delta_i : S_i \to S_i \; \forall i \in N.$$
2 Equilibrium in Transportation Networks

While considering the traffic game, one important issue is to define the strategy set of all the players. In this game, the strategy set would consist of the paths from the source to the destination. However, enumerating all the paths between two nodes in a graph becomes computationally complex. So, we limit the number of paths by one of the following ways.

- Considering the first K shortest paths between the source and destination using Dijkstra’s algorithm.
- Considering only paths that do not go beyond the neighbourhood of the source and destination (i.e) paths that do not deviate from the source or destination.

In this section, we discuss about the equilibria in transportation networks by considering a simple example. Consider the network shown in Fig. 1 (excluding the link $A \rightarrow B$)

Let us assume that there are 4000 users who wish to travel from $START$ to $END$. Now, the process proceeds as follows. For the first user, both the paths are of the same travel time and hence he uses one of those. The second user will now use the path that wasn’t used by the first user. Now, for the
third user, the two paths are of the same travel time and so on. A simple equilibrium will be the case when 2000 users use the path \( START \rightarrow A \rightarrow END \) and the other 2000 players use the path \( START \rightarrow B \rightarrow END \).

### 2.1 Nash Equilibrium Approach

In the aforementioned simple equilibrium, none of the 4000 users will have the incentive to switch paths since it would result in a higher travel time. This distribution is now the Nash equilibrium distribution and the total travel time for every player is the same (65). Thus, computing Nash equilibrium for the given network will intuitively result in the best-response strategies for all the players. In addition, Tim Roughgarden in his paper [4] has proved that the travel time in the Nash equilibrium distribution would be \textit{atmost twice} that of the social optimum and thus establishing an upper bound on the performance of the Nash equilibrium distribution.

#### 2.1.1 Computation

The computation of Nash equilibrium for a \textit{general n-player} game will involve scanning through all possible combinations to find the best strategy of a player against the best strategy of all other players.

#### 2.1.2 Performance

However, there has not been any significantly fast way of computing Nash equilibria of general \( n \)-player games. The running time of existing algorithms are exponential since they compute it using a brute force approach. Also, since the traffic patterns are dynamic in nature, the Nash equilibrium has to be computed periodically and this would require even more computational
2.1.3 Programming

As part of a programming exercise, a program was written to compute the Nash equilibrium of a game generated in \emph{GAMUT} using a brute-force approach and this has been tested for upto 10 players with 4 strategies for each player. However, as the size of the input (number of users or the number of strategies for each player or both) increased, the running time increased.

Hence, although the Nash equilibrium seems to be a good option to alleviate congestion, the computational complexity of computing it makes it less viable. In the Internet or any real-world traffic networks, the number of players and the number of strategies for each player are very large, we must look for an alternative approach such as the correlated equilibrium as discussed below.

2.2 Correlated Equilibrium Approach

Correlated equilibrium proves to be useful in this case because of the following reasons. [1]

- A correlated equilibrium is always guaranteed to exist

- It can be found in polynomial time for any number of players and any number of strategies.

The first reason arises from the fact that Nash equilibrium is a special case of a correlated equilibrium. The second reason is because of the fact that the equilibrium can be computed by linear programming since all the quiescence properties are linear in nature.
Let us consider the game discussed in 1.2. The correlated equilibrium for that game can be written mathematically as follows. [1]

\[\sum_{s \in S_{-p}} [u_{is}^p - u_{js}^p]x_{is} \geq 0 \] (1)

where \(i\) is the strategy profile resulting from \(S \in S_{-p}\) by adding the component \(s \in S_p\).

Intuitively, if a trusted intermediary were to draw a strategy profile \(s\) from this distribution and announce to each player \(p\) separately (and privately) \(p\)s own component \(i\), \(p\) will have no incentive to choose another strategy \(j\) assuming that the other players also conform to the intermediarys suggestion, \(i\) is the optimum response in expectation.

### 2.2.1 Computation

The computation of correlated equilibrium is essentially the same as solving the linear program [1]. It has been shown that *Every game has a correlated equilibrium* by showing that the dual of this linear program is infeasible[1]. Although the *simplex method* is much easier to use to solve the linear program, the *ellipsoid algorithm* is used because of its polynomial running time. (The simplex method has an exponential worst case running time)

The dual program has polynomially many variables and exponential number of constraints. However, for the primal program, it is the other way around and hence the dual is being solved by the ellipsoid algorithm. The algorithm described in [1] makes use of *succinct games*.
A succinct game $G = (I, T, U)$ is defined in terms of a set of efficiently recognizable inputs $I$ and two polynomial algorithms $T$ and $U$. For each $z \in I, T(z)$ returns a type, that is an integer $n \geq 2$ (which is the number of players) and an $n$-tuple $(s_1, s_2, s_3, ....s_n)$ where $n_i$ is the number of strategies available to the player $i$. If $n$ and the $t'_i$s are polynomially bounded in $|z|$, the game is said to be of polynomial type. The algorithm $U$ takes this as input and returns a succinct game.

2.2.2 Performance

This algorithm to find the correlated equilibria consists of the following parts; conversion of the directed graph to a game, obtaining the succinct form of the game, obtaining the linear program that models the correlated equilibrium and finally using the ellipsoid method to solve the linear program. Since each of these steps takes polynomial time, the entire algorithm will run in polynomial time.

2.2.3 Programming

The given input direct graph has to be modelled as a game before it can be converted into a succinct game. For this, as mentioned above, we make use of either Dijkstra’s algorithm to enumerate the first $K$ shortest paths or define a neighbourhood for consideration. The value of $K$ or the size of the neighbourhood depends on the topology of the network.
3 Applications

3.1 Alleviating network congestion

As mentioned above, the main application is to alleviate traffic congestion by obtaining an equilibrium distribution so that no user will have the incentive to deviate from his/her original path unilaterally. Correlated equilibria provide a more natural means to achieve this.

3.1.1 The Internet

Packet routing is one of the most frequent operations performed in the Internet and the problem of congestion occurs very often in routers and access points leading to packet loss. By making use of the correlated equilibrium, each user can be given the route he/she must follow and that would be the best option available to them. Once the congestion decreases, packet drop rate decreases leading to lesser number of failed transmissions.

3.1.2 Real World Traffic Networks

Congestion occurs not just in the Internet, but also in real-world traffic networks. Almost every major city in the world suffers from this problem. This algorithm will go a long way in reducing congestion in these traffic networks and also allow easy management for the transport authorities.
3.2 Braess’s Paradox

Dietrich Braess in [5] had shown an important result in traffic networks called the Braess’s paradox. It can be simply stated as follows.

*Adding a new link to an existing traffic network need not necessarily decrease the travel time. On the contrary, in some cases, it may lead to a redistribution of traffic due to the shift in the Nash equilibrium leading to a reduction in the efficiency of the network and hence an increase in the travel time.*

The reason behind this is as follows. Consider the network in Fig. 1

Initially, the network consists of two routes from START to END which are \( \text{START} \rightarrow A \rightarrow \text{END} \) and \( \text{START} \rightarrow B \rightarrow \text{END} \). Let us consider that there are 4000 cars willing to travel from START to END. The aim is to find the Nash equilibrium distribution of traffic. In this network, if 2000 cars follow the route via \( A \) and 2000 cars follow the route via \( B \), the total travel time will be 65 for both paths. Now, let us consider that a fast link \( A \rightarrow B \) is added to the network and let us say that the travel time for that link is 0. In this case, all users would prefer the path \( \text{START} \rightarrow A \rightarrow B \rightarrow \text{END} \) since the travel time in that case would be

\[
t = \frac{4000}{100} + 0 + \frac{4000}{100} = 80 \tag{2}
\]

But the path \( \text{START} \rightarrow A \rightarrow \text{END} \) or the path \( \text{START} \rightarrow B \rightarrow \text{END} \) both have a travel time

\[
t = 45 + \frac{4000}{100} = 85 \tag{3}
\]
This is a Nash equilibrium distribution since no player in this setup will have the incentive to unilaterally deviate from his/her current strategy as that would increase their time of travel. However, the Nash equilibrium cost in this case is higher than that of the network before the fast link was added. Thus the addition of the new fast link degraded the performance of the traffic network.

3.2.1 The Internet

In the case of internet, the addition of new high speed links can result in the occurrence of Braess’s paradox. This algorithm would be helpful in the analysis of the network so that potential scenarios of Braess’s paradox can be avoided.

3.2.2 Real World Traffic Networks

Similarly, in real world traffic networks, the planning agencies of the government can make use of this algorithm to see if the construction of a new high speed link such as a flyover or a bridge actually reduces congestion or results in the Braess’s paradox thus saving a lot of time, money and effort.

4 Conclusion

The equilibrium concepts in game theory can thus be effectively used for the design of intelligent transportation systems. This, combined with automation, can prove effective for future intelligent transportation systems.
References


[7] Y.Narahari *Lecture Notes on Game Theory* Indian Institute of Science, Bangalore