Congestion Games

Nicholas B. Andrews
Department of Aeronautics and Astronautics
University of Washington
Seattle, USA
nian6018@uw.edu

***Abstract*—Congestion games are a powerful tool for modeling network flow problems that commonly occur in physical systems. The goal of this project is to develop a simulation for calculating equilibria of congestion games. Dijkstra's shortest path algorithm is employed for calculating equilibria. The result of the simulation is an interactive plot that allows for equilibria and routing sensitivity studies as a function of total players. Experimental results indicate that equilibrium cost is not always the same for all players and can plateau as a function of number of players in some instances. Additional simulation improvements and open congestion game research topics with applications to satellite communication and learning in decentralized multi-agent systems are presented.**

I. Introduction

Congestion games are a class of games that are commonly used for modeling communication and traffic networks. In a congestion game players are trying to minimize their cost between 2 points while sharing the same set of possible paths with other players. The focus of this project is to investigate congestion games and to develop a representative simulation for modeling them. The methods used in the simulation and experimental results are discussed in later sections. Finally the project report concludes with proposals on future research topics and their applications to physical systems.

II. Congestion Model

The basis of congestion games is that the cost incurred by a player is a function of the number of other players that choose the same facility. A congestion game C can be formally represented as

$$C(N, M, (\Sigma^i)_{i \in N}, (c_j)_{j \in M}) \tag{1}$$

where

$N =$ number of players
$M =$ set of facilities
$\Sigma^i =$ set of possible strategies for player i, where each $A^i \in \Sigma^i$ is a nonempty set of facilities
$c_j(k) =$ cost to each user of facility j, if there are exactly k users
$\sigma_j(A) =$ number of users of facility j

The total cost v^i accrued by player i for choosing strategy set A is the sum of the cost of each facility in A. The total cost function is defined as $v^i : \Sigma \to R$ where

$$v^i(A) = \sum_{j \in A^i} c_j(\sigma_j(A)) \tag{2}$$

Congestion games are most commonly represented as a network graph where the edges of the graph represent facilities and the nodes represent a junction of a subset of facilities. The objective of each player is to minimize the cost of transit from one node to another. The cost per facility is typically labeled along it's respective edge as shown in the example in figure 1.

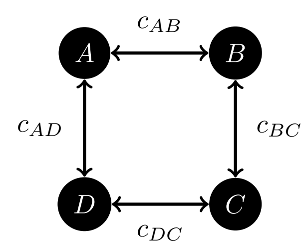

Fig. 1. Congestion game with bidirectional and fixed cost facilities.

There are 2 subtleties to consider when defining a congestion game: the start and destination nodes per player and the directionality of facilities. The start and end nodes across all players could be the same or entirely different per player. Although, games in which all players are travelling to and from the same nodes seem to be studied most often. In addition, facilities can be unidirectional or bidirectional. A unidirectional facility has different cost functions depending on the direction travelled, while a bidirectional facility has the same cost function regardless of direction.

III. Braess' Paradox

A motivating example for studying congestion games is Braess' Paradox. The congestion game is shown in figure 2.

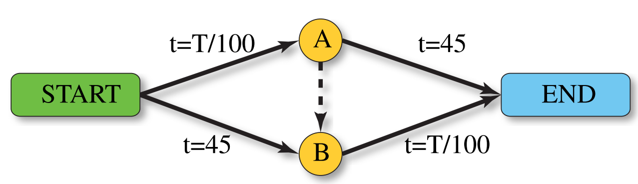

Fig. 2. Braess' Paradox network.

The paradox can be seen through the following example: consider a game in which there are 4000 players that wish to travel from start to end. They can choose between 2 routes: start-A-end and start-B-end. The players achieve an equilibrium travel time of 65 minutes when 2000 players travel each route. Now, consider the addition of a new zero cost facility connecting A-B. The new network has an equilibrium cost of 80 minutes with all players travelling start-A-B-end. This leads to the unintuitive result that the addition of more facilities can lead to larger equilibrium costs.

It has been observed that real road networks and vehicle traffic have demonstrated Braess' Paradox. In some cases where a major roadway was temporarily closed, the overall commute time of travelers was reduced [3]. Congestion games could act as an important tool for modeling traffic and for planning future road construction. For example, it would be essential to know how a new road would affect traffic patterns before spending time and money building the road.

IV. Congestion Game Simulation

A. Development

The primary focus of this project was the development of a modular and scalable congestion game simulation for calculating equilibria. The simulation was developed in Python 3 and utilizes the Matplotlib and NetworkX modules for plotting. The simulation allows the user to define the number of players, nodes, facilities, and cost functions. The simulation currently assumes bidirectional facilities and the same start and end nodes for all players.

In order to calculate the equilibrium of a congestion game the best response strategy had to be developed first. The best response is the minimum cost route from start to end. However, as the size of the congestion game increases it can become computationally intensive to calculate every possible route in order to find the cheapest route. To relieve this computational burden Dijkstra's algorithm was implemented. The details of Dijkstra's algorithm will be discussed in the next section, but in short it is a method for calculating the shortest path between 2 nodes without having to look at every possible permutation.

The equilibrium is calculated by playing the best response dynamics one player at a time and updating the facility cost prior to the next player's response. After all players have played, then the equilibrium cost can be calculated by re-processing and summing the path per player through the facility cost functions. The pseudocode for the equilibrium calculation is shown in algorithm 1.

Dijkstra's algorithm was first validated independently before being integrated with the rest of the simulation. The equilibrium calculation was validated against Braess' paradox and other test cases found online. The entire simulation was tested against a variety of scenarios and several pre-defined scenarios are included with the simulation package.

Algorithm 1 Equilibrium calculation

```
Require: Graph, source, target
  for number of players do
    compute minimum cost path with Dijkstra
    update number of users of facilities along path
  end for
  for number of players do
    cost = 0
    for facility in player path do
      cost + = cost of facility
    end for
    return  total cost for player
  end for
```

B. Dijkstra's Algorithm

Dijkstra's algorithm is is a method for determining the shortest path between nodes on a graph. In a more general application it can be used to find the shortest path between a source node and any other node. However, for it's application to this problem it only needs to find the distance between the start and end nodes. It is commonly used in network routing problems and broadly falls under the category of graph and tree search algorithms.

The fundamental idea behind the algorithm is to search along the node that has the smallest total cost from the source. For example, suppose the source node is connected to 2 nodes A and B. Facility start-A has a cost of 5, and start-B has a cost of 3. The algorithm begins at the start node and adds the neighbors of the start node to a list. Out of that list the node with the smallest cost to the source is selected; B in this case. Node B is removed from the list, and node B's neighbors are added to the list. Once again, the node with the smallest cost to the source is selected and the process is repeated until the end node is added to the list. Pseudocode for Dijkstra's algorithm is presented in algorithm 2.

C. Results

The outputs of the simulation are: cost and route per player, and an interactive plot. The plot includes a histogram of costs per player and a network graph of the game with edge colors representing the number of players choosing the facility. The sliding bar at the top allows the user to change the total number of players in the game. The histogram and network edge colors update dynamically with the total players. The purpose of the sliding bar is to enable an easier analysis of equilibrium cost and number of players per facility as a function of total players.

In the example output provided in figure 3 there are 100 total players. A majority of the players have an equilibrium cost of 190.5, while a few have an equilibrium cost of just above 111. The facility G-end is the most traveled, and is actually used by all 100 players. The facilities with a darker shade of blue, such as C-D, are rarely, if at all, selected.

V. Discussion

After experimenting with several different congestion games, an unexpected observation was that the equilibrium cost of some scenarios plateaued once the total number of players exceeded some threshold. It is unclear if this is a result of the chosen facility cost functions and other assumptions made in the simulation, or some underlying symmetry present in the network. An example of this can be seen in the Braess'

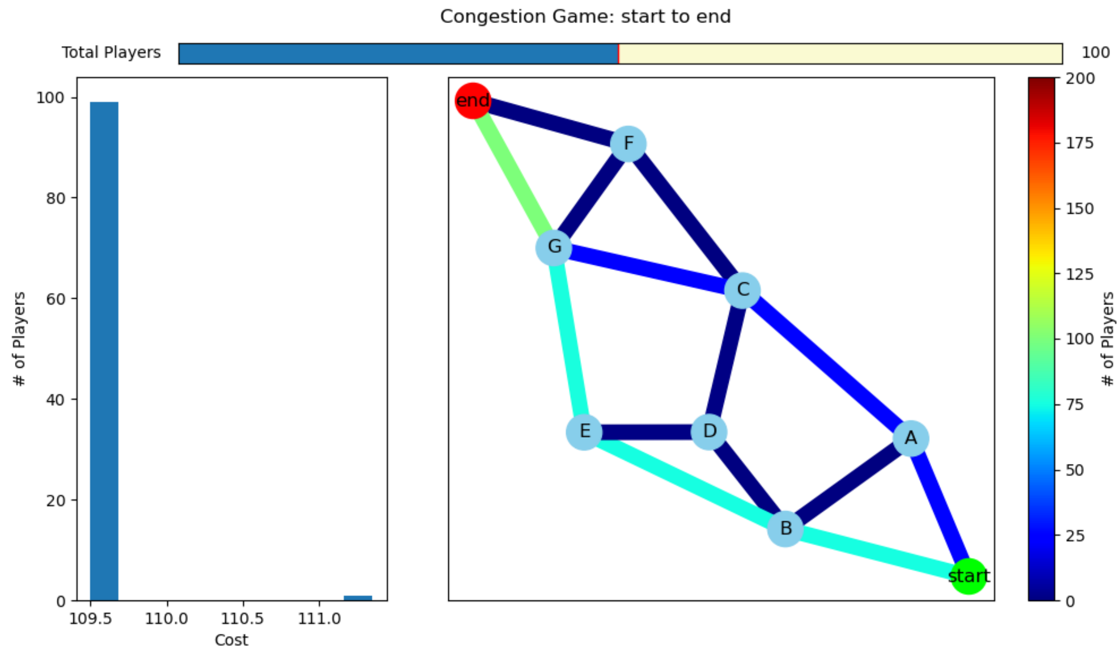

Fig. 3. Output plots from simulation.

Paradox example in figure 2. Once the total number of players is greater than or equal to 4500, the equilibrium cost plateaus at 90.

Another surprising observation was that the equilibrium cost was not always the same for all players. This behavior is likely due to some asymmetry in the network. Without having a specific representative physical system, it was difficult to determine reasonable cost functions. Only linear and fixed facility cost functions were experimented with in the simulation. Although the simulation can handle a cost function of any form, it became difficult to reason about equilibrium results as the cost functions became more complex.

Lastly, it was interesting to see how the number of players per facility changes as a function of total players. Facilities that may be unoccupied with less total players in the game, could become crucial in maintaining equilibrium cost as the total number of players increases. This result is the reverse of Braess' Paradox and is more intuitive: adding additional facilities decreases cost.

VI. Future Work

A next step for the simulation developed in this project would be to apply it to a real world system; such as a communication or road network. An intermediate step to making the jump to a real world system would be to relax the bidirectional and same start and end nodes assumptions. The same start and end nodes is an especially strict and unrealistic constraint. Removing this assumption would open up a whole new dynamic to the congestion game equilibrium behavior and would better reflect how players behave in the real world.

A particular physical system of interest is a satellite communication network, such as the SpaceX Starlink constellation. The interest in this system is due to the fact that the facility cost and number of users per edge varies temporally. However, as mentioned in the previous sections, a fair amount of background knowledge of the physical system would be required in order to develop representative cost functions. It seems like fitting a curve to empirical data of the costs per player would be a valid approach for deriving approximate cost functions without much prerequisite knowledge about the inner workings of the physical system.

An additional future research topic is to apply learning dynamics. This would be most applicable to a congestion network that represents a decentralized system, like communication paths between a swarm of autonomous multi-agent robots, where each agent has to learn the quickest path to send a message from one agent to another without knowing the exact position/facility costs of the other agents. This type of framework is ripe for the bandit, or possibly semi-bandit, feedback problem. Learning dynamics would be irrelevant in something like a traditional communication network, where a centralized leader has perfect knowledge and assigns minimum cost routes to players. For example, it would be silly if users had to choose which cell towers they wanted to communicate through while the telecommunication company has perfect knowledge of the network and player objectives.

A final future research topic under consideration is to perform a sensitivity analysis on the facilities that compose the network. The primary goal of this research would be to identify if there are facilities present in the network that are recreating Braess' Paradox. The results from this research would provide

Algorithm 2 Dijkstra's Algorithm

```
Require: Graph, source, target
  create vertex set Q
  for vertex v in Graph do
    dist[v] ← ∞
    prev[v] ← None
    add v to Q
  end for
  dist[source] ← 0
  while Q is not empty do
    u ← vertex in Q with min dist[u]
    if u = target then
      return dist, prev
    else
      remove u from Q
    end if
    for neighbor v of u do
      alt ← dist[u] + length(u, v)
      if alt < dist[v] then
        dist[v] ← alt
        prev[v] ← u
      end if
    end for
  end while
  S ← empty list
  u ← target
  while u is defined do
    insert u to beginning of S
    u ← prev[u]
  end while
  return S
```

a method for improving equilibrium cost by simply removing existing facilities. A crude initial approach to this problem would be to alternate the removal of a single facility from the network, calculate equilibrium cost, and see if any of the equilibrium costs with a removed facility are less than the original network.

VII. Conclusion

This project investigated the formulation and equilibria of congestion games. A simulation was developed to build, experiment, and calculate equilibria of congestion games. The algorithms implemented in the simulation were presented and experiment observations were discussed. Simulation improvements and open congestion game research problems, with a focus on applications to physical systems, were presented. The simulation code is available on GitHub at https://github.com/nian6018/Congestion-Game.git.

References

[1] Dov Monderer and Lloyd S Shapley. Potential games. Games and economic behavior, 14(1):124–143, 1996.

[2] Learning with bandit feedback in potential games J. Cohen, A. Héliou, and P. Mertikopoulos. In NeurIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.

[3] Kolata, Gina (25 December 1990). "What if They Closed 42d Street and Nobody Noticed?". New York Times. Retrieved 16 March 2021.

[4] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". Numerische Mathematik. 1: 269–271