

Optimizing Urban Traffic Network Design: A Multi-Objective Evolutionary Algorithm Approach.

Josh Bohde, BS in Computer Science

May 3, 2009

Abstract

This is my abstract. It is very wordy. Words words words.

Keywords

Network Design Problem, Evolutionary Algorithm, Multiobjective

Part I

Introduction

The investigator will develop a method for optimizing the design of urban traffic networks. Important to urban traffic networks are the secondary characteristics of the network, such as the cost of the improvements to the network. Previous methods for solving this problem have involved applying weights to the values of these characteristics in ranking possible solutions. The proposed method will instead use a multi-objective evolutionary algorithm (MOEA) to provide a Pareto Optimal set of urban traffic networks. The objectives of this algorithm will be to maximize performance of the network, minimize cost of changes to the network, minimize emissions, and minimize accidents.

The investigator aims to improve upon previous related research in three ways. First, by using a multi-objective approach to optimize with respect to overall network performance, cost, emissions, and traffic safety. Second, by using a traffic simulator in the fitness function in the evolutionary algorithm in order to provide fitnesses that are more applicable to real world situations. Third, scalability testing in order to demonstrate the feasibility of the proposed method in real world applications.

The inclusion of a traffic simulator presents a challenge in the form of high computational costs. This necessitates investigation into representations of road networks and respective genetic operators that generate valid solutions, while minimizing total evaluations.

As urban areas grow, the aging road network is no longer able to handle the amount of traffic generated by the population, resulting in congestion and delays. These delays translate into a direct economic cost which is rising. In study of urban congestion in the United States [7] shows that the congestion costs rose from \$45.5 billion in 1995 to \$78.2 billion in 2005, adjusted for inflation. This congestion also resulted in 2.9 billion gallons of wasted fuel, and 4.2 billion hours in delays [7]. The proposed research aims to limit this congestion, while balancing a variety of limiting factors to the network. The economic burden placed on the tax payers will be taken into account, as well as car emissions and the likelihood of accidents, with the goal of minimizing these. A multi-objective approach will present a set of solutions that do not dominate each other, allowing for city planners to choose the most appropriate one, given their constraints.

Part II

Background

Urban traffic network design is a version of the network design problem (NDP). This problem is maximizing the amount of traffic on a network while minimizing the cost of the network, between a set of nodes representing places traffic can start or end, and links representing connections between nodes. There are two aspects to optimizing the performance of a network, discrete network design problem (DNDP) and continuous network design problem (CNDP). DNDP optimization is done by altering the discrete variables of the network, which is adding and subtracting links between nodes, which corresponds to adding roads and highways. CNDP optimization is achieved by altering the continuous variables, such as traffic light timings, speed limits, and number of lanes.

Because the number of links is exponential with regards to the number of nodes, the search space for DNDP is very large. With the inclusion of CNDP, the search space is even larger. The search space is not simple, as evidenced by Braess's paradox, which shows that adding capacity to a network can decrease the overall performance of the network. In [6], it was shown that for a random graph, the probability of Braess's paradox occurring is high. This implies that when using stochastic algorithms to solve NDP, the search space is complex.

In [5], the authors identify several other factors that affect the quality of the road network for the city inhabitants, and several restrictions for traffic network design. Two objectives presented are the minimization of emissions and accidents. The effects of emissions are lowered as a result of lower speeds and traffic volume, with space between traffic and distance from residential areas also playing a part. Traffic accidents are less numerous with less traffic volume, and are less severe with lower speeds.

Part III

Related Work

Various search algorithms have been used in traffic network design. In [1], the authors improve the Sioux Falls traffic network, a CNDP, using a bi-level programming model and simulated annealing. Fitness was determined by the travel cost between each node in the graph. Genetic algorithms were

shown to be optimize for CNDP in [4], using a case study of network with 4 nodes. This algorithm employed a deterministic user traffic assignment to establish fitness. Genetic algorithms were shown to be comparable to simulated annealing for optimization of a CNDP using a real traffic network [2]. These were also tested against hill climbing and tabu search heuristics.

In [3], the authors presented a new formulation of the network design emphasizing robustness of the produced system in order to account for varying demands between nodes in the graph. This formulation of the problem and the resulting genetic algorithm was shown to produce high quality solutions on example problems.

A MOEA is a type of evolutionary algorithm to optimize for one or more fitness values without reducing them to a single value. As described in [12], one solution dominates another if its respective fitness values are no less than the other's, and at least one value greater than the other's respective value. Instead of a single solution, a set of non-domination solutions are returned, called the Pareto-optimal front.

While there are many different formulations of MOEAs, a common problem of them is to have a large Pareto-optimal front, especially as the number of fitness values increases. The concept of ϵ -dominance was introduced in [11], which limits the Pareto-optimal front by not allowing solutions that dominate each other less than ϵ to be included within the Pareto-optimal front. Conceptually this divides the area of possibly accepted fitness into a grid, only allowing one solution in each grid.

Part IV

Methodology

1 Evolutionary Algorithm

The proposed MOEA is the ϵ -dominance based multi-objective algorithm (ϵ -MOEA), as described in [10]. It was chosen based upon its property of ensuring diversity amongst the Pareto-optimal front, which is an important quality for this problem.

2 Network Representation

The network is represented as a graph: a set of vertices and edges, with each edge connecting two vertices. The addition and deletion of edges from the

network satisfies discrete aspect of the NDP.

To satisfy the continuous aspect, each edge has several properties affecting the performance of the individual edge. These include the shape of the edge, max speed, number of lanes, edge priority, and the direction of the edge. The priority of an edge determines how likely cars traveling on it are going to yield to cars on some intersecting edge. The max speed is the maximum speed that cars are allowed to travel. The number of lanes determines how many cars can be at the same progression on the edge. The direction of the edge determines whether or not the edge road on it is one way. Finally the shape of the edge determines the placement of the edge on the simulated area. This is represented as a set of points, consisting of floating points. The actual path of the road must visit each of these points in order before reaching the end point.

3 Genetic Operators

3.1 Recombination

Recombination of individuals is achieved by through a process analagous to uniform crossover for bit string representations. For each edge in both parents, the edge from the first parent is assigned to a randomly chosen child, with the remaining edge going to the remaining child. For each edge not in both parents, it is randomly assigned to a randomly chosen child. The exact procedure is given in Algorithm 1.

Algorithm 1 Network Recombination

```

for all  $edge \in parent_1\_edges \cup parent_2\_edges$  do
  Pick a random integer  $r$  uniformly from  $[0, 1]$ 
  if  $edge \notin parent_2\_edges$  then
     $child_r\_edges \leftarrow child\_edges \cap edge$ 
  else
     $child_{(1-r)}\_edges \leftarrow child_{(1-r)}\_edges \cap (edge \text{ in } parent_1\_edges)$ 
     $child_r\_edges \leftarrow child_r\_edges \cap (edge \text{ in } parent_2\_edges)$ 
  end if
end for

```

3.2 Mutation

3.2.1 Discrete

Mutation of networks is achieved by randomly adding or deleting edges. For every child generated, it is mutated one time. The exact mutation procedure is described in Algorithm 2.

Algorithm 2 Discrete Mutation

```
 $n \leftarrow$  number of elements in  $nodes$ 
Pick two random integers  $j, k$  uniformly from  $[0...n]$  where  $j \neq k$ 
Generate an edge  $e$  connecting  $nodes_j$  and  $nodes_k$ 
if  $e \notin edges$  then
     $edges \leftarrow edges \cup e$ 
else
     $edges \leftarrow edges - e$ 
end if
```

3.2.2 Continuous

Continuous mutation is done by altering individual edges. The aspects modified are edge shape, number of lanes, lane spread, and edge priority. Edge shape is modified by taking the current edge shape, choosing a point on it, and randomly moving it. The chance to do this is user defined. This algorithm is described in Algorithm 3.

Algorithm 3 Edge Shape Mutation

```
 $shape$  is the ordered set of points making up the shape of the edge.
Pick a random real number  $r$  uniformly from  $[0, 1]$ .
if  $r < chance\_to\_mutate$  then
    Pick a random integer  $n$  uniformly from  $[1, n - 1]$  where  $n$  is the number
    of elements in  $shape$ 
     $H \leftarrow$  the hyperbolic function created by  $shape_n$  and  $shape_{(n+1)}$ 
    Pick a random real number  $t$  uniformly from  $[0, 1]$ .
     $p \leftarrow H(t)$ 
    Pick two random real numbers  $x_r, y_r$  uniformly from  $[0, 1]$ .
     $p \leftarrow p + (x_r * max\_delta\_x, y_r * max\_delta\_y)$ 
    Insert  $p$  into  $shape$  between  $shape_n$  and  $shape_{(n+1)}$ 
end if
```

Algorithm 4 Lane Number Mutation

Pick a random real number r uniformly from $[0, 1]$.
if $r < \textit{chance_to_mutate}$ **then**
 Pick a random integer l uniformly from $[-\textit{max_delta}, \textit{max_delta}]$
 $\textit{lanes} \leftarrow \textit{lanes} + l$
 $\textit{lanes} \leftarrow \max(1, \textit{lanes})$
end if

Algorithm 5 Lane Spread Mutation

Pick a random real number r uniformly from $[0, 1]$.
if $r < \textit{chance_to_mutate}$ **then**
 Pick a random integer s uniformly from $[-1, 1]$
 $\textit{spread} \leftarrow s$
end if

Algorithm 6 Edge Priority Mutation

Pick a random real number r uniformly from $[0, 1]$.
if $r < \textit{chance_to_mutate}$ **then**
 Pick a random integer p uniformly from $[-\textit{max_delta}, \textit{max_delta}]$
 $\textit{priority} \leftarrow \textit{priority} + p$
 $\textit{priority} \leftarrow \max(0, \textit{priority})$
end if

4 Fitness Measure

The fitness is a tuple, consisting of the network performance and cost.

4.1 Network Performance

Common to all of the previous research is the usage of mathematical models to emulate the traffic of the network. While this is useful for determining the theoretical maximum performance of a network, it may not accurately reflect the performance of a real urban network. To better model the performance of a real network, a traffic simulator is proposed to generate the characteristics of the network from which the fitness will be derived. The planned traffic simulator is Simulation of Urban Mobility (SUMO) [8]. SUMO is a microscopic, space-continuous traffic simulator, developed by the Institute of Transport Research at the German Aerospace Centre. Its microscopic model emulates traffic on a per-car basis, while keeping computation time low. This simulator will provide details of network performance, vehicle emissions, and network safety. In order to test the performance in the simulation, the user must supply a flow file, describing a set of flows, composed of a start edge, and ending edge, a beginning and end time, and the number of times to repeat the route. From this, the program can generate specific routes for evaluating the network. Network performance is measured by compiling the edges generated by the algorithm into a traffic network, and running the simulation. From the simulation output, the average time for a car to reach its destination journey will be measured. This is used as the measure for network performance.

4.2 Network Cost

Network cost is the sum of the land cost, building costs, and intersection costs of the network.

4.2.1 Land Cost

Land cost is determined by a supplied to the algorithm, giving the prices for land at a given interval. Using a raytracing algorithm, each edge produces a corresponding inhabitation matrix.

For each edge in the network, a network inhabitation matrix is produced, where each element is the bitwise or of the coresponding elements in each edge inhabitation matrix.

The overall land cost is determined by summing the multiplication of corresponding elements of the cost matrix and network inhabitation matrix.

need algorithms for these

4.2.2 Building Costs

Building costs require two user set parameters, included in the cost file: cost of road per unit per lane and cost of intersection.

Part V

Experimental Setup

Part VI

Results

Part VII

Discussion

Part VIII

Conclusion

Part IX

Future Work

Appendices

A Code Repository

<http://github.com/joshbohde/mondp/>