Project Title: *Simulation and Stochastic Programming Approaches to Accounting for Delay Propagation in Airline Crew Scheduling*

Problem Context and Statement: Given a predetermined schedule of arrivals and departures from a network of airports, an airline must decide how best to assign a set of flights to crew itineraries at minimum cost. This problem, known as the crew-pairing problem, may be formulated deterministically as an integer programming problem:

$$\min \sum_{p \in P} c_p x_p$$

Subject to

$$\sum_{p \in P} \delta_{fp} x_p = 1 \quad \forall f \in F$$

$$x_p \in \{0,1\} \qquad \forall p \in P$$

Where

F	set of all flights
P	set of all feasible pairings
c_p	cost of pairing p
δ_{fp}	1 if flight f is included in pairing p and 0 otherwise

And 0 otherwise

x_p=1 if pairing p is selected in the solution

However, this deterministic formulation does not account for the effects of unexpected delays. A crew arriving on a delayed flight may not be able to leave with their next scheduled flight at the planned time, and this might be especially likely with tight crew schedules. A crew pairing solution that does not account for this uncertainty risks delay propagation across multiple flights. Flight delays cost the airline industry tens of billions of dollars a year ($31 Billion in 2007 [Ball et. al, 2010]), and a significant fraction of these delays are caused by late arriving flights (34% in 2007 [Presentation by Vaze and Wei, 2015]).

To find solutions to the crew pairing problem that are more robust to random delays, we will implement two different modifications to the deterministic crew scheduling problem that incorporate random delay into the model and compare the results from these methods. One approach will be to solve the deterministic crew pairing problem for the top *n* most optimal solutions (*n* to be decided based on feasibility), and then test these *n* crew schedule solutions in simulations that will subject flights to randomly timed delays of random duration (using parameters estimated from flight delay data, see below). Based on the results of these simulations over multiple runs, we will estimate the magnitude and costs of delay propagation, and find the schedule with the minimum total costs (of both crew pairing and delay).

Our second approach will be to include the expected value of the probabilistic delay costs of each pairing in the objective function of the optimization problem, turning the problem into a stochastic integer programing problem. After we get delay data from Bureau of Transportation Statistics (BTS), we will generate a special distribution for the length of delays for each flight, estimate the cost of delays, and solve the optimization problem for the new crew pairing schedule result.

Expected Conclusions:

We will compare the results and solutions from our simulation-based approach and our stochastic programming approach, note any similar conclusions, and make a choice of an optimal crew pairing for our data set.

Data Sources: We will take our data on crew schedules from a regional carrier, ExpressJet. We will estimate our delay parameters from data derived from the BTS Statistics online database.

Algorithms and Software: For the integer and stochastic programming aspects of the project we plan to use CPLEX, and optimization software package with java. For the simulation portion we will use MATLAB.

Problem context and tentative problem statement

We will apply operations research to tic-tac-toe. Specifically, we will compute the optimal strategy for a 3x3 board game and simulate this strategy against various opponents. We will then study connect-3 tic-tac-toe on a 4x4 board, possibly using alpha-beta pruning and a scoring heuristic to make the search tractable.

Models/algorithms/software

The game will be modeled by a game tree where each node is a game state (9-tupe for 3x3) and the descendents are the ≤ 9 possible successor states. As player take turns, each level of the tree (subsequent successor state) alternates between an "x" and an "o" being placed.

To compute the optimal strategy, we will enumerate the game tree to the leaves and then compute a strategy which maximizes expected outcome. The expectation can be taken assuming a random opponent, or if the opponent is optimal alpha-beta pruning can be used.

For 4x4 tic-tac-toe, we foresee difficulty in enumerating all O(16!) game states in the game tree. Thus, we plan on using alpha-beta pruning to reduce the search space and a scoring heuristic to truncate the search tree and make the computation feasible. Some scoring heuristics we are considering are:

- # of ways to win in 1 move + # ways to win in 2 moves
- Max # own adjacent squares - max # opponent adjacent squares

To simulate, we plan on using this open souce MATLAB tic-tac-toe simulation code. Some opponent strategies we are considering to test our computed strategies against are:

- "Randomized": Pick next move at random
- "Reasonable": If can win, win. If going to loose, block. Otherwise, random.

Expected Conclusions

It is known that the optimal tic-tac-toe strategy never loses. Thus, one conclusion of our project will be to validate this computationally. Additionally, we expect to make conclusions about the winning percentages a random player and a player following the "reasonable" strategy will have against an optimal player.

In order to confirm the validity of our optimal strategy, we would simulate the game in two ways. First, we would simulate a situation where both players are playing optimally. If we our strategy is correct, we would hope to observe that the outcome of each game is a draw. Second, we would simulate the game where one player is playing optimally and the other is playing at random. As discussed earlier, we would expect to observe that the player playing optimally would either win or draw every time.

Investigating 4x4 tic-tac-toe will yield conclusions about how well each heuristic performs in this new context, shed insights on how much more difficult 4x4 tic-tac-toe is, and possibly even provide an existence proof (or disproof) for optimal strategies in 4x4 tic-tac-toe.

- **Introduction of Dark Pool**

A dark pool market is a securities trading forum with no published information regarding what is happening in the market. There is complete lack of transparency in a dark pool market. Dark pools came about primarily to facilitate block trading(A large quantity of securities) by institutional traders. Each player in the dark pool market will provide the IOIs(Indication of interests) which may contain any of the following information: the ticker of the stock, the buy/sell price or how much is up for grabs. However, all of these IOIs will not be public to any other player in this market.

- **Why dark pool (Advantages and drawbacks)**

Pros:

1. Reduced market impact
2. Lower transaction costs

Cons:

1. Pool participants may not get the best price
2. Vulnerability to predatory trading by HFTs(High frequency trading)
3. Small average trade size reduces need for dark pools

- **Three Types of Dark Pool**

There are three types of dark pool markets based their trading mechanism: 1. Matching at their exchange price: This form of dark pools matches customer orders at prices derived from lit venue, such as the midpoint of the national best bid and offer (NBBO) or the volume-weighted average price (VWAP). [Zhu, MIT] This type of mechanism does not provide direct price discovery, therefore it is not the mechanism we will focus on in this project. 2. Non-display limit order books: Non-display Limit Order Books provides limited price discovery, This type The price of his form of dark pool is bounded between National Best Bid and Offer(NBBO). 3. Electronic market makers: The third type of dark pool functions as an electronic market maker which accepts and rejects customer orders at high speed. The price is not calculated using NBBO, rather it is determined by price discovery. Therefore, this is the type of dark pool we are going to focus on.

- **Problem Statement**

We would like to simulate the third type of dark pool using the simulation method and try to get the best pricing model for each of the players in the dark pool(the number of players starts from two) by applying the game theory.

- **Tentative Plan**

We are going to collect stock price data from NYSE and Nasdaq to benchmark and determine a tentative range of the bid and sell prices. Since this price discovery mechanism requires matching between bidders and sellers, we are going to utilize methods/tools in game theory to determine their trade offs and maximize individual's stock value. By the end of this project, we expect to successfully simulate the third type of dark pool as indicated previously, and test several bidding/selling strategies to determine the optimal approach under certain assumptions and restrictions.

Problem Context/Statement:

Supply constraints are limiting the electricity production from natural gas in the northeast. Natural gas demand in the northeast is increasing and pipeline infrastructure is being built out in the northeast. We are looking into how the existing network will respond to uncertainty and changes in capacity. Additionally, we want to run a flow maximization analysis using the existing network.

Tentative Plan:

The data is readily available from the Energy Information Administration and we have already collected the data. We have inflow and outflow capacities from every state in the US and Canada. We are planning to use a flow maximization and run a simulation of the demand variation. We expect to discover the states which the Northeast is dependent upon for natural gas based electricity and where we need to add additional capacity to make the network more stable.

Problem Statement:

- Almost all supply chains have multiple members which typically include a supplier who supplies the raw material or the basic components, a manufacturer who makes the actual product, a warehouse/distributor, and a retailer
- Coping with variable demand is a challenge every supply chain faces. While some industries are able to predict demand with high accuracy, a lot of companies still face uncertainty in their demand.
- In such situations, the bullwhip effect comes into play which distorts demand greatly, leading to overproduction or stock-outs, which might both prove very costly for all members of the supply chain
- Our aim is to model a simple four member supply chain (supplier-manufacturer-distributor-retailer) and try to understand the strategies that different members of the supply chain should adopt in order to be efficient in their operations
- Most companies take a myopic view of their operations and try to maximize profits for their organization alone, frequently at a cost to other members in the supply chain
- We are looking to model a game to see if a broader view of success might help the whole system benefit more (for example, by giving up a small percentage of their profits, the manufacturer might be able to increase profits for their supplier greatly. This will reduce the risk of losing the supplier and improving operations due to better relationship with the supplier, both of which are results with tangible benefits in the long run)

Data and software:

1. We intend to use data from academic cases (such as the Barilla case) wherever possible. We expect that this will not be sufficient by itself, so apart from data we can find on the internet, we will also use fictitious data that follows a certain distribution (eg. normal distribution to represent seasonal demand), and use that to approximate real life demand.
2. We expect to use optimization sub-problems in games and network optimization in this project. MATLAB will be our primary software, and we will also use R and Minitab if necessary.
3. We hope to arrive at optimal supply chain strategies which are not readily intuitive for all members of the supply chain, for different demand distributions. This carries a great practical significance since demand forecasting is what drives most businesses till date. There are still a lot of companies that operate with sub-optimal production planning and absorb the costs of inaccurate forecasting. We hope to shed some light on how to approach uncertain and/or variable demand such that the entire supply chain benefits and minimizes losses.

Problem Context:
With hospital facing increasing pressure on cost reduction, Appointment Scheduling (AS) is becoming more and more important in outpatient services. Long patient waiting time will increase the percentage of no-shows, while long doctor idle time will cause great cost. Effective AS system regulates the flow of patients so that both patient waiting times and doctor idle times are minimized.

Tentative Plan:

1. Data Sources:
   Collect from public healthcare data bases like NAMCS to get the data in patient appointment, check-in and check-out time statistics, etc. Or collect data from local clinics and hospitals.

2. Models/Algorithms/Software:
   This project is based on optimization, queuing theory and discrete-event simulation. Simulation will run in Matlab.
   The preliminary plan is to use queuing theory to analyze a defined AS system, then use the discrete-event simulation model to reveal insights on the effect of changing different parameters, such as appointment interval, unpunctuality of patients, lateness of doctors, patient walk-in and no-show rates.
   Based on the results, I will design an optimized AS rule and get better simulated results compared to collected data.

3. Expected Conclusion:
   In the end, the project is able to use simulation to evaluate the performance of alternative AS and understand the relationship between various environmental factors and various performance measures. In the defined AS system, the optimization of maximizing net profit for health providers is able to be achieved.