**Team 1 Optimization Project Proposal**

**Bar Crawl Optimization**

"What care I how time advances? I am drinking ale today." - Edgar Allen Poe

For over six thousand years, beer has been engrained in our culture. Dating back to nearly 5000 BC, ancient civilizations brewed varieties of beer in large quantities. The beer was used for sustenance, medical purposes and as a form of currency. Recipes and brewing techniques developed over time. In 2018, the U.S. beer industry sold over 200 million barrels of beer[1]. Based on beer shipments and the US Census, U.S. citizens of age 21 years and older consumed 26.5 gallons of beer per person in 2018[1]. To say we love our beer would be considered an understatement by many.

Given this affinity for beer, many people find themselves indulging in the activity of drinking beer on the weekends in the form of a "bar crawl". However, there is an ever-growing number of potential bars to go to, making it difficult for a group to decide on the best set of bars to include on the bar crawl. Team 1 is here to help!

We propose creating a model for our final project that will optimize a group's bar crawl route to optimize a function including the aggregate rating of all of the bars that are included on their crawl and the total walking plus wait time at bars needed to complete the crawl. Using a Yelp dataset for the various cities in North America, we will employ a combination of an assignment model and the Traveling Salesman algorithm to select a set of bars that the group will include on their bar crawl, including the suggested route between the bars. Using inputs from the user such as the number of bars they would like to visit, the time and day they want to do the crawl, their desired walking time, and specific parameters regarding the quality and quantity of the Yelp reviews for the bars, we will create additional constraints for the model to ensure the user's desires are met.

**Exhibit A** below outlines a formulation of the linear program that will be used to solve the problem, which would be an integer program.

**Exhibit B** below outlines an initial view of the pareto optimal solution set that we will provide to the users so that they can choose the bar crawl that is best aligned with their preferences.

---

[1] Americas Beer Distributors: Industry Fast Facts https://www.nbwa.org/resources/industry-fast-facts

**Exhibit A: Bar Crawl Linear Program Formulation**
**Parameters and Sets**

We will define the following set for the formulation of our linear program:

- $LOCATIONS = \{1, \dots, n\}$, where n is the number of locations included in our data set
- $DAYS = \{\text{Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}\}$
- $S = $ Potential subtours of $LOCATIONS$, $\forall S \subseteq LOCATIONS$

The following parameters will also be used in our formulation:

- $d_{i,j}$ = Walking time (seconds) from location i to location j $\quad \forall i, j \in LOCATIONS, i \neq j$
- $s_i$ = Yelp star rating of location i $\quad \forall i \in LOCATIONS$
- $r_i$ = Number of Yelp reviews for location i $\quad \forall i \in LOCATIONS$
- $o_{i,k}$ = Open time for location i on day k $\quad \forall i \in LOCATIONS, \forall k \in DAYS$
- $c_{i,k}$ = Close time for location i on day k $\quad \forall i \in LOCATIONS, \forall k \in DAYS$
- $w_i$ = Average wait time at location i $\quad \forall i \in LOCATIONS, \forall k \in DAYS$
- $b_i$ = Expected time spent at location i $\quad \forall i \in LOCATIONS, \forall k \in DAYS$

We will also ask the user for the following inputs to formulate the relevant constraints for our model:

- $STOPS$ = Number of desired locations to visit during the bar crawl
- $AREA$ = Desired area for bar crawl (latitude and longitude coordinates of central point)
- $START$ = Start time of bar crawl
- $END$ = End time of bar crawl
- $DAY$ = Day of week for bar crawl
- $T_{total}$ = Max time willing to walk from location to location during bar crawl
- $T_{between}$ = Max time willing to walk between any two locations during bar crawl
- $REVIEWS$ = Minimum number of reviews the user requires for a location to be included
- $STARS$ = Minimum star rating the user requires for a location to be included

**Decision variables**

We will calculate the optimal bar crawl route with the following decision variables:

- $Y_i = \begin{cases} 1, \text{if location i is chosen to be included on the route} \\ 0, \text{otherwise} \end{cases} \quad \forall i \in LOCATIONS$
- $X_{i,j} = \begin{cases} 1, \text{if route from location i to location j is included on the crawl} \\ 0, \text{otherwise} \end{cases} \quad \forall i \in LOCATIONS$

**Objective Function**

For a given distance walked, the objective function will maximize a weighted average of the aggregate Yelp star rating of the bars, walking time to the bars, and wait time at the bars included on the crawl:

$$\max \sum_{i \in LOCATIONS} \alpha * (Y_i * s_i) - \sum_{i,j \in LOCATIONS} (1-\alpha) * X_{i,j} * d_{i,j} - \sum_{i \in LOCATIONS} (1-\alpha) * (Y_i * w_i)$$

$where\ \alpha\ is\ a\ parameter\ within\ the\ range\ (0,1)$

**Constraints**

*Inbound and Outbound Arcs*

As outlined in the Traveling Salesman Problem, each location that is included on the bar crawl must have 1 incoming edge and 1 outgoing edge to ensure the bar crawl enters and leaves each location exactly once:

$$\sum_{j\,\in LOCATIONS, j\neq i} X_{i,j} = Y_j, \qquad \forall j \in LOCATIONS$$

$$\sum_{i\,\in LOCATIONS, i\neq j} X_{i,j} = Y_i, \qquad \forall i \in LOCATIONS$$

We will include an initial node that will be the "starting" and "ending" point of the bar crawl such that the crawl will not have to return to the first bar. The distance from the initial node to all other bars will be set to 0 so that it will have no impact on the distance constraints that we have within the model.

*Subtours*

A Subtour constraint is included to ensure that the number of edges between the nodes within *S* is at most the number of nodes within *S* minus 1:

$$\sum_{i,j\,\in S, j\neq i} X_{i,j} \leq |S| - 1, \forall S \subseteq LOCATIONS \; and \; S \neq \emptyset$$

*Locations visited*

The number of locations visited on the bar crawl must be equal to the number of locations specified by the user:

$$\sum_{i\in LOCATIONS} Y_i = STOPS$$

*Max time between locations*

The maximum walk time between any 2 locations on the bar crawl must be less than or equal to the maximum time specified by the user:

$$X_{i,j} * d_{i,j} \leq T_{between}, \forall i, j \in LOCATIONS$$

*Max total time*

The maximum total walk time for the full bar crawl must be less than or equal to the maximum time specified by the user:

$$\sum_{i\in LOCATIONS} \sum_{j\,\in LOCATIONS, j\neq i} X_{i,j} * d_{i,j} \leq T_{total}$$

*Rating minimum*

The rating for each location on the bar crawl much be greater than or equal to the minimum rating specified by the user:

$$Y_i * s_i - Y_i * STARS \geq 0, \forall i \in LOCATIONS$$

*Review minimum*

The number of reviews for each location on the bar crawl much be greater than or equal to the minimum rating specified by the user:

$$Y_i * r - Y_i * REVIEWS \geq 0, \forall i \; \in LOCATIONS$$

*Open time*

The opening time of any location included on the bar crawl on the specified day must be before the time that the bar crawl will visit the bar. In other words, the start time of the crawl plus the sum of the walking time, wait times, and time spent at and between each bar up until bar *i* needs to be at least the open time of bar *i*.

$$START + \sum_{j=1}^{i-1} (w_j + b_j + d_{i-1,i}) + d_{i-1,i} + w_i \geq o_{i,k} * Y_i$$
$$\forall i \; \in LOCATIONS \; (ordered) \text{and } k = DAY \text{ as specified by user}$$

*Close time*

The closing time of any location included on the bar crawl on the specified day must be after the bar crawl departs the location. In other words, the start time of the crawl plus the sum of the walking time, wait times, and time spent at and between each bar up until and including bar *i* needs to be at most the close time of bar *i*.

$$START + \sum_{j=1}^{i} (w_j + b_j + d_{j-1,j}) \; \leq \; c_{i,k} * Y_i$$
$$\forall i \; \in LOCATIONS \; (ordered) \text{ and } k = DAY \text{ as specified by user}$$

*Front-loading the distance*

A safer solution would involve as little walking as possible between the last bars. As such, we introduce a new constraint such that the distance walked in the second half of the crawl should be at most 80% of the distance in the first half. If the number of walks is uneven (i.e. the number of bars is even), the middle walk is not counted in the averages.

$$0.8 * \sum_{j=1}^{\lfloor i/2 \rfloor} d_j - \sum_{j=\lceil i/2 \rceil + 1}^{i} d_j \geq \; 0$$

**Exhibit B: Giving users multiple options**

Although we maximize the user rating for a given walking distance, we will give the user the option to choose the best combination of distance walked vs average rating through a Pareto-like efficiency bar graph.

As such, users can choose directly their desired tradeoff of distance and rating in the user interface and obtain the recommended list of bars.