Abstract light trails in white, orange, and blue on a black background, resembling quantum paths or data streams.

Quantum Impact Collective: Optimization of Food Resources



In the U.S., fast food restaurants
produce 22 to 33 billion pounds of
food waste by annually.

Goal: To optimize vehicle routing from shelters to collect food resources in Boston.

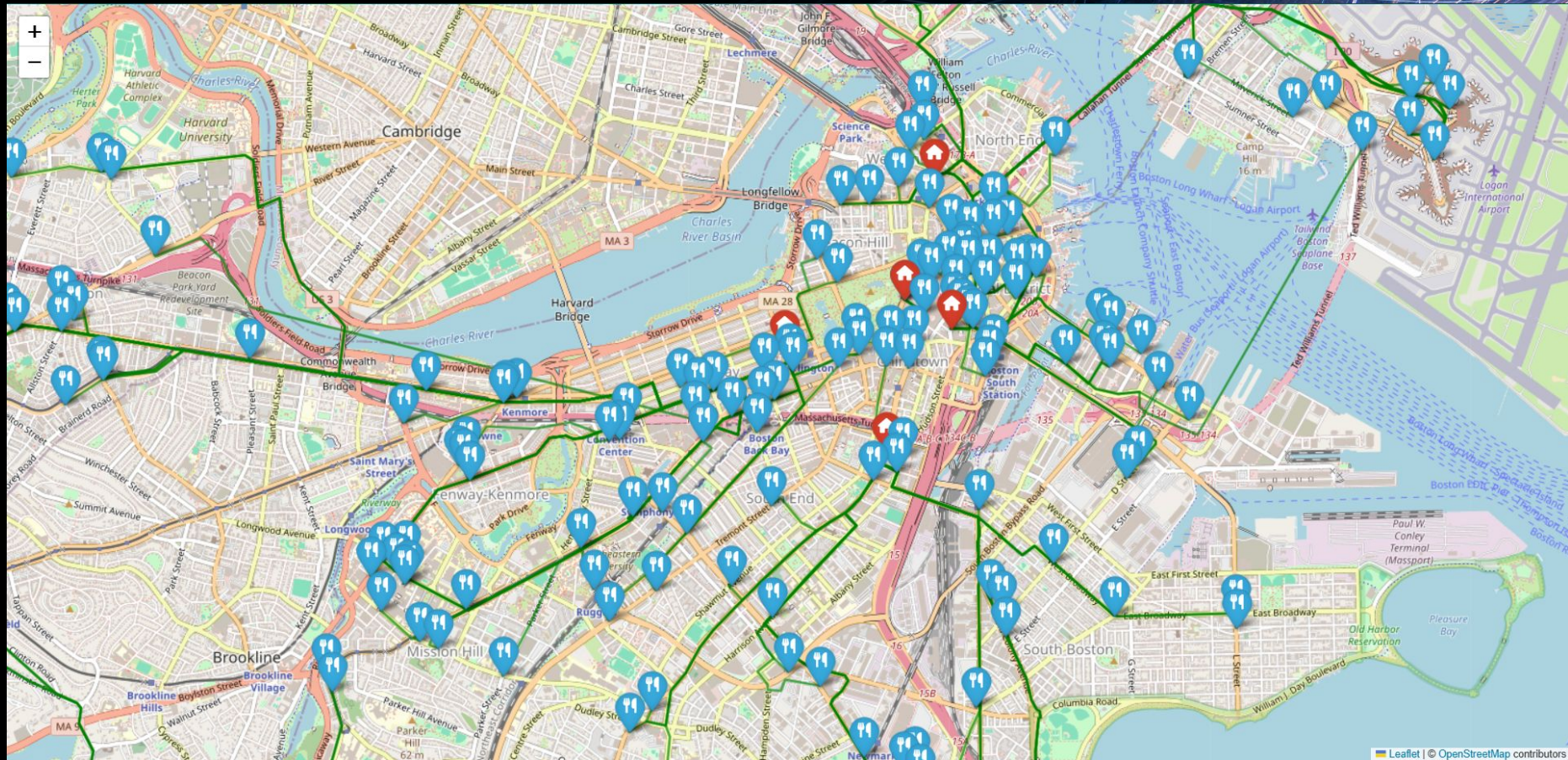
Shapiro, "Fast Food's Contribution to Food Waste," *Shapiro*, Mar.
08, 2024. <https://shapiroe.com/blog/junk-food-waste/>

Process

1. Find Homeless Shelters in Boston
2. Determine fast food locations in Boston
3. Correlate locations between shelters and fast food locations
 - a. Identify proximity based on longitude and latitude
 - b. For simplicity purposes, we chose the closest 3 restaurants to each shelter
4. Establish a situation

1	Shelter	Latitude	Longitude
2	Family-Aid Boston	42.2981	-71.1166
3	Bridge Over Troubled Waters	42.35535	-71.063
4	Boston Rescue Mission	42.35375	-71.0595
5	Boston Night Center	42.36262	-71.06067
6	Pine Street Inn	42.3458	-71.06463

1	businessname	latitude	longitude
2	APPLEBEE'S NEIGHBORHOOD GRILL & BAR	42.32608686677121	-71.06361196234216
3	Auntie Anne's Pretzels/ Carvel	42.35195868500647	-71.05502724058353
4	Buffalo Wild Wings Go/Sp. AS2-A18	42.364539333151285	-71.02181778282163
5	Burger King	34.244386753001244	-73.6513909034731
6	Burger King	42.309265153704665	-71.05775317571394
7	Burger King	42.35631361369498	-71.06192094875507
8	Burger King	42.36883243461128	-71.03932826829823
9	Burger King	42.35301796551194	-71.13497876705965
10	Burger King	42.33921949208055	-71.05095898205067
11	Burger King	42.26820347573987	-71.09577982327994
12	Burger King	42.262078075396296	-71.1085691449198
13	Burger King	42.3861663166383	-71.00953086689843
14	Burger King # 3483	42.26588887601696	-71.16795211651815
15	Burger King # 3531	42.27597672024672	-71.13916232653567
16	CALIFORNIA PIZZA KITCHEN	42.34715955305909	-71.08251052626302
17	Cava	42.35312267843186	-71.05742343463301



Establish a Situation

Consider the following situation for simplicity:

Suppose we have 3 **shelters** (Depos)

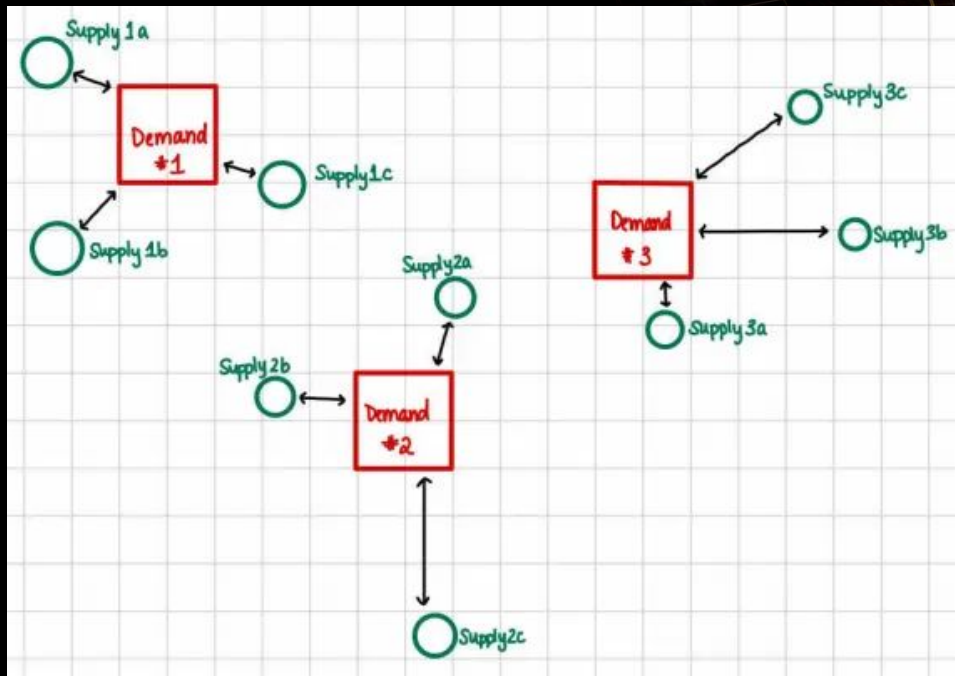
→ Each shelter has an **X set demand**

Based on distance, we have 3 **restaurants** for each shelter (Clients)

→ Each restaurant has a **Y set supply**

→ Restaurants are located at varying distances from each restaurant, but are the closest in terms of Euclidean distance.

How can we create an optimal route that not only takes the **shortest distance**, but also **completes the X demand** needed for the shelter?





Capacitated Vehicle Routing

Definition: An optimization that involves finding the best routes for a fleet of vehicles to deliver items to customers.

Characteristics:

- Each vehicles has a maximum capacity
- Each vehicles starts and ends at the depot (shelter)
- Each client can only be served by 1 vehicles (restaurant)

Modifications to fit our problem:

1. Clients have supplies, depots have demand
2. Constraints include fulfilling demand (factor for optimization)

We attempted using a nonlinear model, a **quadratic model**, using the LeadHybridNSampler

Process of Application

1. Consider how to apply constraints to Quadratic Model → ConstrainedQuadraticModel
 - a. PROBLEM : CVR uses a DiscreteQuadraticModel (DQM) – different methods
 2. Enforce constraints on model using penalties → BinaryQuadraticModel
 - a. PROBLEM: penalties added too much constraint
 3. Utilize CVRP, but create separate depo + demand variables → 3 Arrays instead of 2
 - a. PROBLEM: CVRP is defined in an interesting way
 - i. The demand object MUST be an array, where the first element represents the demand of the depo which MUST be 0
 - ii. The client array MUST have the first index represent the depo
- * Using the current model of capacitated_vehicles_routing, we CANNOT define our situation.
- ** HOWEVER, we can present an alternative case, if we had editing access to CVR

Alternate Method

We attempted to treat the depo as its own variables and the depo's demand as an integer value.

Rather than populating an array to be [200, 0, 0, 0], which is invalid according to the current CVR model, we can simply:

- Take a route and track distance traveled and whether demand was met

- After all routes are exhausted, compare values are run the optimization algorithm to determine the least costly route in terms of both distance and demand met.

```
1 from dwave.optimization.generators import capacitated_vehicle_routing
2 from dwave.system import LeapHybridNLSampler
3
4 # Define the shelter and demand
5 depot = (0, 0)
6 depot_demand = 200 # Depot has demand
7
8 # Define the surrounding restaurants and their associated supply
9 restaurants = [(15, 38, 50), (23, -19, 100), (44, 62, 100)]
10
11 # Extract locations and supplies
12 locations_x = [x for x, y, s in restaurants]
13 locations_y = [y for x, y, s in restaurants]
14 supply = [s for x, y, s in restaurants]
15
16 # Initialize the demand list with depot demand and zero for restaurant demands
17 demand = [depot_demand] + [0] * len(supply) # Depot has demand, restaurants have no demand
18
19 # Ensure depot is included in locations
20 locations_x = [depot[0]] + locations_x # Add depot coordinates to the list
21 locations_y = [depot[1]] + locations_y # Add depot coordinates to the list
22
23 # Create the CVRP model to fulfill demands + minimize distance
24 model = capacitated_vehicle_routing(
25     demand=demand, # Demand list where depot has non-zero demand, others are zero
26     number_of_vehicles=3, # Updated to 3 vehicles
27     vehicle_capacity=200,
28     locations_x=locations_x,
29     locations_y=locations_y,
30     depot_x=depot # Explicitly provide depot_
31 )
32
33 # Track the demand fulfillment
34 num_samples = model.states.size() # This is important to determine how many samples were returned
35 for i in range(min(3, num_samples)):
36     print(f"Objective value {int(model.objective.state(i))} for")
37
38 # Initialize variables to track the supply usage
39 demand_met = False
40 remaining_demand = depot_demand # Track remaining depot demand dynamically
41
42 # Check the routes and calculate if the demand has been met
43 for j, r in enumerate(routes):
44     print(f"\t Route {j + 1}: {r.state(1)}")
45
46     # Subtract the supply from the current demand based on the route
47     for restaurant_idx, supply_value in enumerate(supply):
48         if restaurant_idx == r.state(i):
49             remaining_demand -= supply_value
50
51     if remaining_demand <= 0:
52         demand_met = True
53         break # Stop processing if demand is met
54
55 # Check if the demand was met
56 if demand_met:
57     print(f"Demand successfully met after route {i + 1}.\n")
58 else:
59     print(f"Demand not fully met after route {i + 1}.\n")
60
```

The background of the slide is a dark blue field filled with a complex network of thin, glowing lines. These lines are primarily blue and white, with some orange and red lines interspersed, particularly on the right side. The lines form a dense, web-like pattern that suggests a network or a complex system. The overall effect is a high-tech, digital aesthetic.

Now Consider:
Grocery Stores → Depo
Shelters → Clients

** Abides by the constraints set up by the CVR

Food Deserts

Food Deserts are defined as regions that don't have access to "healthy" food. These areas often show signs of severe health issues, such as asthma and high obesity. We can provide options by involving grocery stores with healthy food. We can apply this optimization problem by creating the following situation:

Grocery Stores → Depos
Communities/Regions → Clients



Results

```
snelters.append((row[1],row[2]))
route:  <_cython_3_0_11.generator object at 0x16c86a8c0>
Objective value 331 for
    Route 1: [3. 1. 2.]      Route 2: [0. 4.]
    Feasible: True
Objective value 331 for
    Route 1: [3. 1. 2.]      Route 2: [4. 0.]
    Feasible: True
Objective value 331 for
    Route 1: [2. 1. 3.]      Route 2: [4. 0.]
    Feasible: True
> (.venv) (base) druhibhargava@Druhis-MacBook-Pro quantum_hack %
```


Further Steps

- Quadratic Assignment Problem (QAP)
- Distribution of other resources
 - Mobile Food Markets
 - University food waste

QAP

Considering the previous problem, where:

Shelters \rightarrow Depos

Restaurants \rightarrow Clients

In a more realistic situation, we have trucks with larger capacities than the shelters; thus, they gain an excess of food. If that is the case, excess food at shelters is still considered “waste”. Thus, following the defined QAP protocol, we can transport this “waste” from shelter to shelter at certain times during the month. We can then consider additional costs of moving from shelter to shelter.

