Sub-Team 2:

Modeling Travel-Related Sustainability Objectives: A Multi-Objective Approach

Spring 2024

Iliyan • Karan • Michelle • Alex

Rohan • Tejas • Yeongbin

Mentor: Kibria

CREATING THE NEXT®

Problem Addressed

- Limitations of traditional traffic assignment objectives
- How to ensure mobility, access, safety, and equity?
- Single-objective vs multi-objective

Objectives

- Study multiobjective optimization basics
- Model accessibility and safety equity
- Extend single objective model to multiobjective
- Coding in Python

Literature review

- All members

Study Optimization Solvers

- All members

Model accessibility equity

- Tejas, Karan, Rohan, Michelle, Alex

Coding

- All members

Model safety equity

- Tejas, Yeongbin, Iliyan

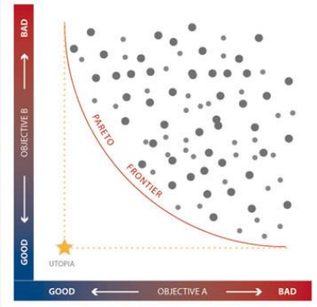
Tasks

Learn about different optimization solvers

Model accessibility equity

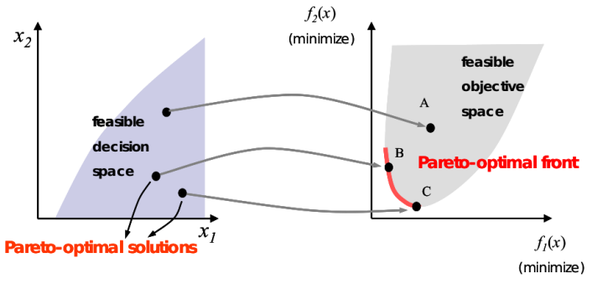
Model safety equity

Formulate multiobjective model



Pyomo coding and implementation

Study multiobjective optimization (MOO) basics



Activities and Work Completed

Learn about different optimization solvers

GLPK: Linear program (LP) solver

COIN|OR

CPLEX: Linear, quadratic, mixed-integer, and mixed-integer quadratic program solver

IBM CPLEX

COMPUTATIONAL INFRASTRUCTURE FOR OPERATIONS RESEARCH — COIN|OR

IPOPT: Large-scale nonlinear program solver

COUENNE: Non-convex mixed-integer nonlinear program solver

Activities and Work Completed

Model accessibility equity

- Consider opportunities in each zone
- Model equitable access to opportunities

Types of accessibility function:

- Gravity-based accessibility
- Distance-decay functions
- Cumulative opportunities

$$A_r = \sum_{s \in \mathcal{Z} \setminus \{r\}} \frac{\sum_{k \in \mathcal{K}^{rs}} O_s e^{-T_k^{rs}}}{q_{rs}}$$

$$\min \sum_{r \in \mathcal{Z}} \sum_{s \in \mathcal{Z} \setminus \{r\}} |A_r - A_s|$$

Activities and Work Completed

Model safety equity

- Estimate number of accidents
- Consider severity of accidents

$$\lambda_r = \sum_{s \in \mathcal{Z} \setminus \{r\}} \frac{\sum_a \sum_{k \in \mathcal{K}^{rs}} S_a(x_a)^\theta \delta_{a,k}^{(r,s)}}{q^{rs}}$$

$$\min \sum_{r \in \mathcal{Z}} \sum_{s \in \mathcal{Z} \setminus \{r\}} |\lambda_r - \lambda_s|$$

Formulate MOO model

$$\min \sum_{(p,q) \in \mathcal{D} \setminus \{(r,s)\}} \left| \frac{\left(\sum_k \frac{T_k^{rs}}{L_k^{rs}}\right)}{|\mathcal{K}^{rs}|} - \frac{\left(\sum_k \frac{T_k^{pq}}{L_k^{rs}}\right)}{|\mathcal{K}^{pq}|} \right|$$

$$\min \sum_{(p,q) \in \mathcal{D} \setminus \{(r,s)\}} |A_r - A_s|$$

$$\min \sum_{(p,q) \in \mathcal{D} \setminus \{(r,s)\}} |\lambda_r - \lambda_s|$$

Subject to:

$$\sum_{k \in \mathcal{K}^{(r,s)}} f_k^{(r,s)} = q^{rs}, \forall (r,s) \in \mathcal{D}$$

$$x_a = \sum_{(r,s) \in \mathcal{D}} \sum_{k \in \mathcal{K}^{(r,s)}} f_k^{(r,s)} \delta_{a,k}^{(r,s)}, \forall a \in \mathcal{A}$$

$$T_k^{(r,s)} = \sum_a \left(t_a^0 (1 + x_a^2)\right) \delta_{a,k}^{(r,s)}, \forall k \in \mathcal{K}^{(r,s)}, \forall (r,s) \in \mathcal{D}$$

$$f_k^{(r,s)} \geq 0, \forall k \in \mathcal{K}^{(r,s)}, \forall (r,s) \in \mathcal{D}$$

Activities and Work Completed

Pyomo coding

- Code organization
- Use of *dictionaries* and *for* loops
- Solver call

Parameters

Decision variables

Objectives

Constraints

```python
#model
def setup_accessibility_model(nodes, od_demand, opportunities, travel_time):
    model = pyo.ConcreteModel()

    # Define the sets based on the inputs
    model.nodes = pyo.Set(initialize=nodes)
    model.od_pairs = pyo.Set(dimen=2, initialize=od_demand.keys())

    # Parameters
    model.opportunities = pyo.Param(model.nodes, initialize=opportunities)
    model.od_demand = pyo.Param(model.od_pairs, initialize=od_demand)
    model.travel_time = pyo.Param(model.od_pairs, initialize=travel_time)

    # Decision variables for accessibility measure
    model.accessibility = pyo.Var(model.nodes, within=NonNegativeReals)

    # Objective: Minimize the sum of absolute differences in accessibility between all pairs of nodes
    def objective_rule(m):
        return sum(abs(m.accessibility[r] - m.accessibility[s])
                   for r in m.nodes for s in m.nodes if r != s)
    model.objective = pyo.Objective(rule=objective_rule, sense=pyo.minimize)

    # Constraint: Definition of accessibility measure for each node
    def accessibility_rule(m, r):
        return m.accessibility[r] == sum(m.opportunities[s] * pyo.exp(-m.travel_time[r,s])
                                         / m.od_demand[r,s] for s in m.nodes if r != s)
    model.accessibility_constraint = pyo.Constraint(model.nodes, rule=accessibility_rule)

    return model
```

```python
model = setup_accessibility_model(nodes, od_demand, opportunities, travel_time)

opt = SolverFactory('gurobi')
results = opt.solve(model)

for node in model.nodes:
    print('Accessibility of node {}: {}'.format(node, pyo.value(model.accessibility[node])))

print('Objective value (equity in accessibility):', pyo.value(model.objective))
```

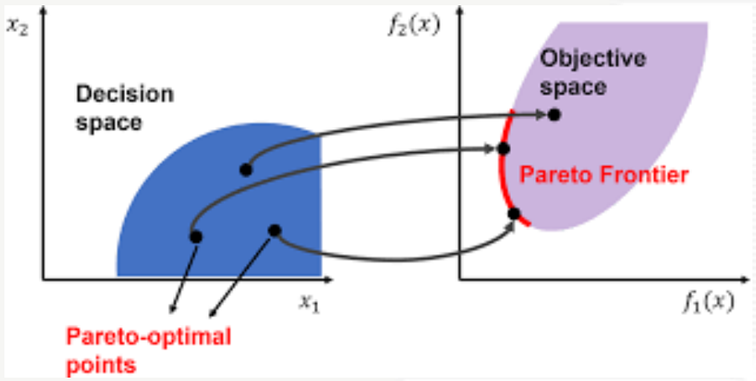
Solver call

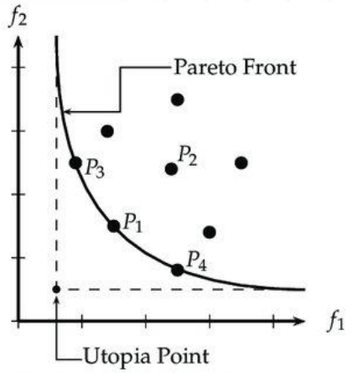
Activities and Work Completed

Study MOO

Basics

- Decision space vs feasible criterion space
- Pareto solution and Utopia solution
- Pareto frontier





Solution methods

- Weighted-sum
- Weighted min-max
- Lexicographic

$$U = \sum_{i=1}^{k} w_i f_i(\mathbf{x}) \qquad U = \max_i \left\{ w_i \left[f_i(\mathbf{x}) - f_i^{\circ} \right] \right\}$$

Minimize (for $i = 1$ to k):

$$f_i(\mathbf{x})$$

subject to:

$$f_j(\mathbf{x}) \le f_j(\mathbf{x}_j^*); \quad j = 1 \text{ to } (i-1); \quad i > 1; \quad i = 1 \text{ to } k$$

Challenges

Modeling related

- Modeling accessibility function
- Modeling safety function
- Nonlinearity and nonconvexity

Coding related

- Unprocessed data
- Link and node attributes
- Problem size

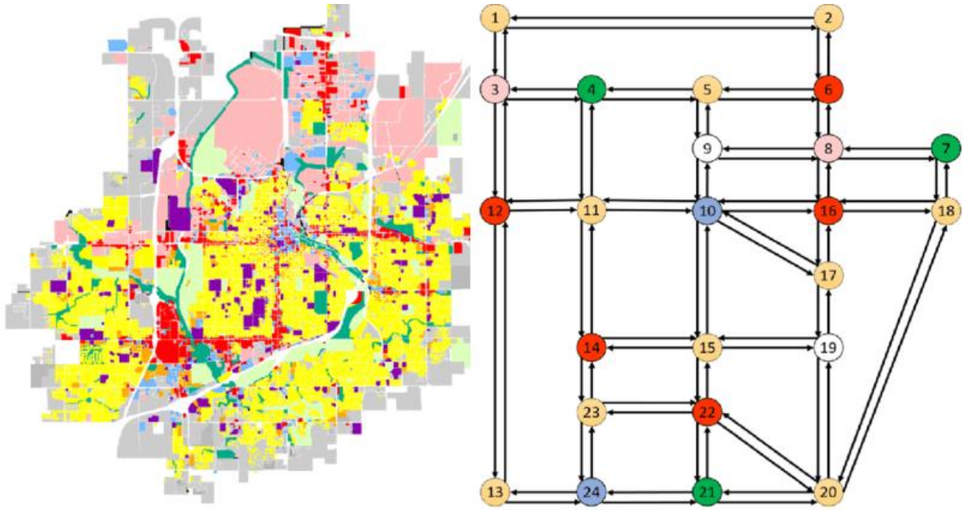
Lessons Learned/Skills Obtained

Learned optimization problem formulation basics

$$\min_{x \in R^n} f(x)$$
$$s.t. g(x) \leq 0$$
$$h(x) = 0$$
$$x_L \leq x \leq x_U$$

Familiarity with Pyomo and different solvers for optimization tasks

Gained insights into multiobjective traffic assignment and network modeling



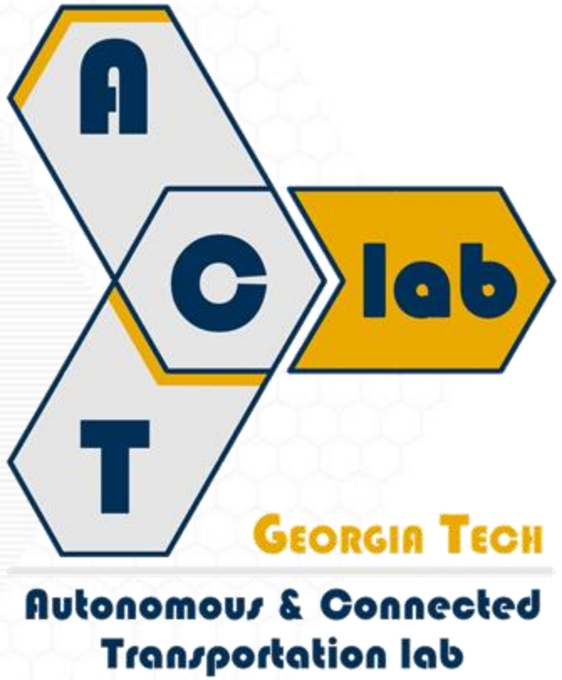
Learned concise coding for different problem size

Learned to implemented weighted-sum method for MOO

$$U = \sum_{i=1}^{k} w_i f_i(\mathbf{x})$$

Next Steps

- Finish modeling safety equity
- Model parameter calculations
- Solving the multiobjective model
  - Explore algorithms
  - Explore solvers


A
C
lab
T
Georgia Tech
Autonomous & Connected
Transportation lab

Questions?