Model Formulation

Model 1: Fixed Assignment - Trucks are assigned to specific districts and cannot move between districts. Trucks operate on each day of the week in either the morning or evening timeslots.
Model 2: Shared Trucks - Trucks can move between districts across days, allowing shared trucks between districts. Trucks also operate on each day of the week in the specified timeslots.
Each model will consider each day of the week, indexed from 1 to 7. Trucks will be assigned to one of two available timeslots on each day, allowing us to model pickups across the entire week.

Sets and Parameters

- D: Set of districts, indexed by d (12 districts in total).
- T: Set of trucks, indexed by t (total of 2,230 trucks).
- C: Capacity of each truck in tons (e.g., 12 tons per truck).
- $W_{d,d'}$: Daily waste generated in district d on day d', in tons (randomly generated).
- h: Time taken to collect one ton of waste.
- SEI_d: Socio-economic indicator for district d.
- f_d: Required pickup frequency in district d (2–3 times per week).
- $t_{d,d'}$: Travel time between districts d and d'.
- α, γ, β: Weights for efficiency, equity, and maximizing garbage collection.
- $Days = \{1, 2, \ldots, 7\}$: Days of the week.
- $TimeSlots = \{\text{Morning}, \text{Evening}\}$: Two available timeslots per day.
- k: Index for timeslots (Morning = 7am–8am, Evening = 6pm–7pm).
- $MaxTrucks = 2230$: Maximum total number of trucks available across all districts.

Decision Variables

- $x_{d,t,k,d'} \in \{0, 1\}$: Binary variable indicating if truck t is assigned to district d during timeslot k on day d'.
- $y_{d,d',t,d''} \in \{0, 1\}$: Binary variable indicating if truck t moves from district d to district d' on day d'' (used in the shared-trucks model).
- $g_{d,d'}$: Total amount of garbage collected in district d on day d' over the week.

Objective Function

The objective function is designed to:

- Minimize the total collection time.
- Minimize the equity gap in garbage collection.
- Maximize the total garbage collected.

The objective function is:

$$\min \quad \alpha \cdot \text{TCT} + \gamma \cdot \text{PGC} - \beta \cdot \text{TGC}$$

where:

$$\text{TCT} = \sum_{d \in D} \sum_{t \in T} \sum_{k \in TimeSlots} \sum_{d' \in Days} x_{d,t,k,d'} \cdot \left(h \cdot \frac{W_{d,d'}}{C} \right) + \sum_{d \neq d'} \sum_{t \in T} \sum_{d'' \in Days} y_{d,d',t,d''} \cdot t_{d,d'}$$

$$\text{PGC} = \sum_{d \in D} \left| \sum_{d' \in Days} g_{d,d'} - \sum_{d' \in Days} W_{d,d'} \right|$$

$$\text{TGC} = \sum_{d \in D} \sum_{d' \in Days} g_{d,d'}$$

Constraints

Truck Capacity Constraint: Each truck can only collect up to its maximum capacity in each timeslot on each day.

$$\sum_{d \in D} x_{d,t,k,d'} \cdot W_{d,d'} \leq C, \quad \forall t \in T, \quad k \in TimeSlots, \quad d' \in Days$$

Pickup Frequency Constraint: Each district must have pickups 2–3 times a week.

$$\sum_{t \in T} \sum_{k \in TimeSlots} \sum_{d' \in Days} x_{d,t,k,d'} \geq f_d, \quad \forall d \in D$$

Garbage Collection Constraint: The total garbage collected in each district should match the daily waste generated on each day.

$$g_{d,d'} = \sum_{t \in T} \sum_{k \in TimeSlots} x_{d,t,k,d'} \cdot W_{d,d'}, \quad \forall d \in D, \quad d' \in Days$$

Travel Constraints (for Shared Trucks Model): Trucks can travel between districts if shared across districts on the same day.

$$y_{d,d',t,d''} = 1 \Rightarrow x_{d,t,k,d''} = 1 \quad \text{and} \quad x_{d',t,k,d''} = 1, \quad \forall d, d' \in D, \quad t \in T, \quad k \in TimeSlots, \quad d'' \in Days$$

Socio-Economic Priority Constraint (Equity): Prioritize pickups in districts with high socio-economic indicator values to ensure equitable service.

$$f_d \cdot W_{d,d'} \cdot SEI_d \approx \text{constant across} \, d \in D$$

Total Truck Availability Constraint: The total number of trucks assigned across all districts in each timeslot must not exceed the total available fleet (2,230 trucks).

$$\sum_{d \in D} \sum_{t \in T} \sum_{k \in TimeSlots} \sum_{d' \in Days} x_{d,t,k,d'} \leq 2230$$

Non-Negativity and Binary Constraints:

$$x_{d,t,k,d'} \in \{0,1\}, \quad y_{d,d',t,d''} \in \{0,1\}, \quad g_{d,d'} \geq 0$$

Explanation of the Two Models

Fixed Assignment Model: In this model, trucks are pre-assigned to districts and cannot move between them, making $y_{d,d',t,d''} = 0$ for all $d \neq d'$ and for all times and days. Each truck is limited to its assigned district on each day of the week. This results in a simpler model with less flexibility but could help optimize resource allocation within districts.
Shared Trucks Model: Trucks are allowed to move between districts, represented by $y_{d,d',t,d''}$ for shared trucks across different districts on the same or different days. This model provides more flexibility and potentially improves coverage in high-demand districts but introduces additional complexity with cross-district logistics and scheduling.

0.1 The example python code for the two models:

```
import pulp as lp
import random

# Parameters
districts = range(12)  # 12 districts
days = range(7)  # Days in a week
timeslots = ["Morning", "Evening"]  # Two timeslots
total_trucks = 2230  # Total number of trucks available
truck_capacity = 12  # Each truck's capacity in tons
truck_collection_time_per_ton = 1  # Collection time per ton (for simplicity)
alpha = 1.0  # Weight for TCT
gamma = 1.0  # Weight for PGC
beta = 1.0  # Weight for TGC

# Randomly generated data
waste_generated = {d: {day: random.uniform(5, 20) for day in days} for d in districts}  # Random daily waste
generation per district
```

```
frequency_required = {d: random.choice([2, 3]) for d in districts}  # Required pickup frequency
SEI = {d: random.uniform(0.5, 2.0) for d in districts}  # Socio-economic indicator

# Problem initialization
model_fixed = lp.LpProblem("Fixed_Assignment_Model", lp.LpMinimize)
model_shared = lp.LpProblem("Shared_Trucks_Model", lp.LpMinimize)

# Decision variables
x_fixed = lp.LpVariable.dicts("x_fixed", ((d, t, k, day) for d in districts for t in range(total_trucks) for k
in timeslots for day in days), 0, 1, lp.LpBinary)
x_shared = lp.LpVariable.dicts("x_shared", ((d, t, k, day) for d in districts for t in range(total_trucks)
for k in timeslots for day in days), 0, 1, lp.LpBinary)
y_shared = lp.LpVariable.dicts("y_shared", ((d1, d2, t, day) for d1 in districts for d2 in districts if
d1 != d2 for t in range(total_trucks) for day in days), 0, 1, lp.LpBinary)
g_fixed = lp.LpVariable.dicts("g_fixed", ((d, day) for d in districts for day in days), 0)
g_shared = lp.LpVariable.dicts("g_shared", ((d, day) for d in districts for day in days), 0)

# Objective functions
# Total Collection Time (TCT), Proportional Garbage Collection (PGC), Total Garbage Collected (TGC)

def TCT(x):
    return lp.lpSum(x[(d, t, k, day)] * truck_collection_time_per_ton * waste_generated[d][day] /
    truck_capacity for d in districts for t in range(total_trucks) for k in timeslots for day in days)

def PGC(g):
    return lp.lpSum(abs(g[(d, day)] / waste_generated[d][day] - 1) for d in districts for day in days)

def TGC(g):
    return lp.lpSum(g[(d, day)] for d in districts for day in days)

# Define objectives for fixed and shared models
model_fixed += alpha * TCT(x_fixed) + gamma * PGC(g_fixed) - beta * TGC(g_fixed)
model_shared += alpha * TCT(x_shared) + gamma * PGC(g_shared) - beta * TGC(g_shared)

# Constraints
for d in districts:
    for day in days:
        # Fixed assignment model: trucks stay within district boundaries
        model_fixed += g_fixed[(d, day)] == lp.lpSum(x_fixed[(d, t, k, day)] * waste_generated[d][day] for
        t in range(total_trucks) for k in timeslots), f"GarbageCollection_Fixed_{d}_{day}"
        # Shared trucks model: trucks can be shared across districts
        model_shared += g_shared[(d, day)] == lp.lpSum(x_shared[(d, t, k, day)] * waste_generated[d][day] for
        t in range(total_trucks) for k in timeslots), f"GarbageCollection_Shared_{d}_{day}"

        # Pickup frequency constraints
        model_fixed += lp.lpSum(x_fixed[(d, t, k, day)] for t in range(total_trucks) for k in timeslots)
        >= frequency_required[d], f"PickupFrequency_Fixed_{d}_{day}"
        model_shared += lp.lpSum(x_shared[(d, t, k, day)] for t in range(total_trucks) for k in timeslots)
        >= frequency_required[d], f"PickupFrequency_Shared_{d}_{day}"

        # Socio-economic indicator constraints (equity)
        model_fixed += (frequency_required[d] / (waste_generated[d][day] * SEI[d])) >= 0.5  # Example equity
        constraint
        model_shared += (frequency_required[d] / (waste_generated[d][day] * SEI[d])) >= 0.5

# Truck capacity constraint across both models
for t in range(total_trucks):
    for k in timeslots:
        for day in days:
            model_fixed += lp.lpSum(x_fixed[(d, t, k, day)] * waste_generated[d][day] for d in districts)
```

```
            <= truck_capacity
            model_shared += lp.lpSum(x_shared[(d, t, k, day)] * waste_generated[d][day] for d in districts)
            <= truck_capacity

# Total number of trucks available constraint
model_fixed += lp.lpSum(x_fixed[(d, t, k, day)] for d in districts for t in range(total_trucks) for k in
timeslots for day in days) <= total_trucks
model_shared += lp.lpSum(x_shared[(d, t, k, day)] for d in districts for t in range(total_trucks) for k in
timeslots for day in days) <= total_trucks

# Solve the models
model_fixed.solve()
model_shared.solve()

# Display results
print("Fixed Assignment Model:")
print("Status:", lp.LpStatus[model_fixed.status])
print("Objective Value:", lp.value(model_fixed.objective))

print("\nShared Trucks Model:")
print("Status:", lp.LpStatus[model_shared.status])
print("Objective Value:", lp.value(model_shared.objective))
```