V

Individual Assignment 2025

Student Name
Student ID
Course Name

April 21, 2025

1 Problem 1. Transportation Network Optimization

1.1 Problem Description

We consider a three-layer transportation network with components that consists of plants (P), warehouses (W), and customers (C). This system is broken down into two stages of transportation:

- From plants to warehouses;
- From warehouses to customers.

Each plant $i \in P$ has a supply limit s_i, each warehouse $j \in W$ has a capacity c_j, and each customer $k \in C$ has a demand d_k.

The objective is to minimize the total transportation cost.

1.1.1 Part (a): Multi-Source Fulfillment

Customers can receive their demand from multiple warehouses. The constraints ensure supply, capacity, and demand are respected.

1.1.2 Part (b): Single-Warehouse Assignment

Each customer must be assigned to only one warehouse. This adds a binary decision variable and changes the model slightly.

1.2 Model

The mathematical model of this task consists of the next input parameters:

1.2.1 Parameters and notions

- p: positive integer - number of plants;
- w: positive integer - number of warehouses;

- c: positive integer - number of customers;
- A set of plants $P = 1..p$
- A set of warehouses $W = 1..w$
- A set of customers $C = 1..c$
- s_i: supply limit - how many items can be produced on i-th plant;
- c_j: capacity of warehouse $j \in W$;
- d_k: demand of client $k \in C$
- a_{ij}: unit costs for shipping from plants to warehouses, $i \in P, j \in W$
- b_{jk} unit costs for shipping from warehouses to customers, $j \in W, k \in C$

1.2.2 Decision Variables

We need to use next additional variables to set up the model:

- $x_{ij} \geq 0$: amount shipped from plant i to warehouse j
- $y_{jk} \geq 0$: amount shipped from warehouse j to customer k

Objective function minimizes total transportation costs in the network, that is, the sum of the transport costs from the plants to the warehouses and the transport costs:

$$\min \sum_{i \in P} \sum_{j \in W} a_{ij} x_{ij} + \sum_{j \in W} \sum_{k \in C} b_{jk} y_{jk}$$

1.2.3 Multisource fulfillment model

Therefore, for task 1a (Multisource fulfillment), we have the following problem:

$$\begin{aligned}
\min \quad & \sum_{i \in P} \sum_{j \in W} a_{ij} x_{ij} + \sum_{j \in W} \sum_{k \in C} b_{jk} y_{jk} \text{ (Objective function)} && (1) \\
\text{s.t.} \quad & \sum_{j \in W} x_{ij} \leq s_i \quad \forall i \in P \quad \text{(Plant supply limit)} && (2) \\
& \sum_{k \in C} y_{jk} \leq c_j \quad \forall j \in W \quad \text{(Warehouse capacity)} && (3) \\
& \sum_{j \in W} y_{jk} \geq d_k \quad \forall k \in C \quad \text{(Customer demand)} && (4) \\
& \sum_{i \in P} x_{ij} \geq \sum_{k \in C} y_{jk} \quad \forall j \in W \quad \text{(Warehouse flow balance)} && (5)
\end{aligned}$$

The first constraint describes that amount shipped from plant is less than its supply limit.

The second constraint describes that goods shipped to warehouse is less warehouse capacity.

The third constraint describes that goods shipped to customers should satisfy their demands.

The forth constraint shows that goods shipped to the customer cannot be greater than those delivered from plants.

1.2.4 Single-Warehouse Assignment Model

The Single-Warehouse assignment model adds the next variables.

- $z_{jk} \in \{0, 1\}$: 1 if customer k is assigned to warehouse j, 0 otherwise

The modified constraints are

$$\min \quad \sum_{i\in P}\sum_{j\in W} a_{ij}x_{ij} + \sum_{j\in W}\sum_{k\in C} b_{jk}y_{jk} \text{ (Objective function)} \tag{6}$$

$$\text{s.t.} \quad \sum_{j\in W} x_{ij} \leq s_i \quad \forall i \in P \quad \text{(Plant supply limit)} \tag{7}$$

$$\sum_{k\in C} y_{jk} \leq c_j \quad \forall j \in W \quad \text{(Warehouse capacity)} \tag{8}$$

$$\sum_{j\in W} y_{jk} \geq d_k \quad \forall k \in C \quad \text{(Customer demand)} \tag{9}$$

$$\sum_{i\in P} x_{ij} \geq \sum_{k\in C} y_{jk} \quad \forall j \in W \quad \text{(Warehouse flow balance)} \tag{10}$$

$$\sum_{j\in W} z_{jk} = 1 \quad \forall k \in C \quad \text{(Each customer is assigned to exactly one warehouse)} \tag{11}$$

$$y_{jk} \leq d_k z_{jk} \quad \forall j \in W, \forall k \in C \quad \text{(Link shipment to assignment)} \tag{12}$$

The objective and constraints (plant supply, warehouse capacity, flow balance) remain unchanged, however added next constraints:

- Each customer is assigned to exactly one warehouse;
- Demand of the client satisfied only by one warehouse.

1.3 Numerical answers

1.3.1 Task 1a

The optimal objective function for instance 1 is:

Objective value: 370.0
Ship 20.0 from Plant 0 to Warehouse 0
Ship 40.0 from Plant 1 to Warehouse 0
Ship 60.0 from Plant 1 to Warehouse 1
Ship 20.0 from Warehouse 0 to Customer 1
Ship 40.0 from Warehouse 0 to Customer 2
Ship 30.0 from Warehouse 1 to Customer 0
Ship 20.0 from Warehouse 1 to Customer 1
Ship 10.0 from Warehouse 1 to Customer 3

The optimal objective function for instance 2 is: objective value: 1275.0
Ship 55.0 from Plant 0 to Warehouse 0
Ship 45.0 from Plant 0 to Warehouse 1
Ship 55.0 from Plant 1 to Warehouse 1
Ship 45.0 from Plant 1 to Warehouse 3
Ship 30.0 from Plant 2 to Warehouse 1
Ship 50.0 from Plant 3 to Warehouse 2
Ship 20.0 from Plant 4 to Warehouse 2
Ship 45.0 from Plant 5 to Warehouse 0
Ship 10.0 from Plant 5 to Warehouse 2
Ship 40.0 from Plant 5 to Warehouse 4
Ship 5.0 from Warehouse 0 to Customer 1
Ship 15.0 from Warehouse 0 to Customer 2
Ship 10.0 from Warehouse 0 to Customer 7
Ship 10.0 from Warehouse 0 to Customer 8
Ship 10.0 from Warehouse 0 to Customer 9
Ship 20.0 from Warehouse 0 to Customer 10
Ship 20.0 from Warehouse 0 to Customer 16
Ship 10.0 from Warehouse 0 to Customer 19
Ship 5.0 from Warehouse 1 to Customer 3
Ship 20.0 from Warehouse 1 to Customer 5
Ship 10.0 from Warehouse 1 to Customer 11
Ship 15.0 from Warehouse 1 to Customer 14
Ship 50.0 from Warehouse 1 to Customer 17
Ship 30.0 from Warehouse 1 to Customer 18
Ship 10.0 from Warehouse 2 to Customer 0
Ship 15.0 from Warehouse 2 to Customer 1
Ship 5.0 from Warehouse 2 to Customer 3
Ship 20.0 from Warehouse 2 to Customer 6
Ship 15.0 from Warehouse 2 to Customer 12
Ship 15.0 from Warehouse 2 to Customer 13
Ship 45.0 from Warehouse 3 to Customer 2
Ship 20.0 from Warehouse 4 to Customer 4
Ship 20.0 from Warehouse 4 to Customer 15

1.3.2 Task 1b

Task 1b: The optimal objective function for instance 1 is: Objective value: 460.0 Ship 10.0 from Plant 0 to Warehouse 2
Ship 40.0 from Plant 1 to Warehouse 0
Ship 40.0 from Plant 1 to Warehouse 1
Ship 30.0 from Plant 2 to Warehouse 2
Ship 40.0 from Warehouse 0 to Customer 1
Ship 30.0 from Warehouse 1 to Customer 0
Ship 10.0 from Warehouse 1 to Customer 3
Ship 40.0 from Warehouse 2 to Customer 2

The optimal objective function for instance 2 is: Objective value: 1275.0
Ship 95.0 from Plant 0 to Warehouse 0
Ship 100.0 from Plant 1 to Warehouse 1
Ship 30.0 from Plant 2 to Warehouse 3
Ship 30.0 from Plant 3 to Warehouse 1
Ship 20.0 from Plant 3 to Warehouse 2
Ship 5.0 from Plant 4 to Warehouse 2
Ship 15.0 from Plant 4 to Warehouse 3
Ship 5.0 from Plant 5 to Warehouse 0
Ship 55.0 from Plant 5 to Warehouse 2
Ship 40.0 from Plant 5 to Warehouse 4
Ship 20.0 from Warehouse 0 to Customer 1
Ship 60.0 from Warehouse 0 to Customer 2
Ship 10.0 from Warehouse 0 to Customer 8
Ship 10.0 from Warehouse 0 to Customer 19
Ship 10.0 from Warehouse 1 to Customer 3
Ship 20.0 from Warehouse 1 to Customer 4
Ship 20.0 from Warehouse 1 to Customer 5
Ship 50.0 from Warehouse 1 to Customer 17
Ship 30.0 from Warehouse 1 to Customer 18
Ship 20.0 from Warehouse 2 to Customer 6
Ship 10.0 from Warehouse 2 to Customer 7
Ship 20.0 from Warehouse 2 to Customer 10
Ship 15.0 from Warehouse 2 to Customer 12
Ship 15.0 from Warehouse 2 to Customer 13
Ship 10.0 from Warehouse 3 to Customer 0
Ship 15.0 from Warehouse 3 to Customer 14
Ship 20.0 from Warehouse 3 to Customer 16
Ship 10.0 from Warehouse 4 to Customer 9
Ship 10.0 from Warehouse 4 to Customer 11
Ship 20.0 from Warehouse 4 to Customer 15

1.4 Disclosures

2 Problem 2. Scheduling with Release Dates and Penalties

2.1 Problem Description

We need to complete a certain number of jobs, each with a processing time, due date, and release date. These jobs are executed by machines, and a job cannot start before its release date. It is not strictly necessary to finish a job by its due date, but a penalty is applied for each unit of time a job finishes late.

This is the classic scheduling problem. Each job must be assigned to exactly one machine, and no machine can process more than one job at a time. Jobs are non-preemptive.

2.1.1 Part (a): Maximize the Number of Scheduled Jobs

Ignore both the penalty and the due date – only the release date matters. The goal is to maximize the number of jobs that can be scheduled while respecting their release dates.

So in this particular formulation we have to achieve the minimal date when all the jobs are done.

2.1.2 Part (b): Minimize Total Weighted Tardiness

We should consider the due dates and penalties. The objective is to minimize the total penalty across all jobs for finishing them after their due dates.

2.2 Model for task 2a

2.2.1 Parameters:

We have the following:

- n jobs, indexed by i or $j = 1..n$, set $J = 1..n$;
- m identical machines, indexed by $k = 1..m$, set $A = 1..m$;
- p_i – processing time for each job;
- r_i – release date (job cannot start before r_i) for each job $i = 1..n$;
- d_i – due date for each job (we should try to do the job before this date) $i = 1..n$;
- w_i – penalty per unit time of tardiness (lateness) for each job $i = 1..n$.

2.2.2 Decision Variables

For the task 2a we define the following variables:

- $s_j \geq 0$: start time of job j;
- $a_{ik} \in \{0, 1\}$: 1 if job i is assigned to machine k, 0 otherwise;
- $y_{ij} \in \{0, 1\}$: 1 if job i started earlier than job j, 0 otherwise;

- T - time, when all the jobs are done.

Model described as following, where big-M is a large enough number:

$$\min \quad T \quad \text{(Objective function)} \tag{13}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} a_{ik} = 1 \quad \forall i \in J \quad \text{(Every job to one machine)} \tag{14}$$

$$s_i \geq r_i \quad \forall i \in J \quad \text{(Release date)} \tag{15}$$

$$y_{ij} + y_{ji} = 1 \quad \forall i, j \in J, i \neq j \quad \text{(Order consistency)} \tag{16}$$

$$y_{ij}(s_j - s_i) \geq 0 \quad \forall i, j \in J, i \neq j \quad \text{(Order defined)} \tag{17}$$

$$(s_i + p_i) \leq s_j + M * (1 - a_{ik} * a_{jk}) + M * (1 - y_{i,j}), \forall i, j \in J, i \neq j \forall k \in A \quad \text{(No overlap)} \tag{18}$$

$$s_i + p_i \leq T \quad \forall i \in J \quad \text{(Finishing date)} \tag{19}$$

Objective function shows that we minimize the time to get all the items.
First constraint marks that each job is assigned to one machine.
Second constraint applied to release dates.
Third condition defines consistency of the order, that i-th job is either earlier or not late than job j.
Forth condition on order defines the order.
Fifth describes that there is no overlapping on same machine.
The last one defines T as the minimal time of completing all the jobs.

2.3 Model for task 2b

For this item we should consider the due dates and penalties. The objective is to minimize the total penalty across all jobs for finishing them after their due dates.

For the task 2b we define the following variables:

- $s_j \geq 0$: start time of job j;
- $a_{ik} \in \{0, 1\}$: 1 if job i is assigned to machine k, 0 otherwise;
- $y_{ij} \in \{0, 1\}$: 1 if job i started earlier than job j, 0 otherwise;
- $T_j \geq 0$: tardiness of job j, defined as $T_j = \max(0, s_j + p_j - d_j)$.

Model described as following, where big-M is a large enough number:

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} w_j \cdot T_j (\text{Objective function}) && (20)\\
\text{s.t.} \quad & \sum_{k=1}^{m} a_{ik} = 1 \quad \forall i \in J (\text{Every job to one machine}) && (21)\\
& s_i \geq r_i \quad \forall i \in J \quad (\text{Release date}) && (22)\\
& y_{ij} + y_{ji} = 1 \quad \forall i,j \in J, i \neq j \quad (\text{Order consistency}) && (23)\\
& y_{ij}(s_j - s_i) \geq 0 \quad \forall i,j \in J, i \neq j \quad (\text{Order defined}) && (24)\\
& (s_i + p_i) \leq s_j + M * (1 - a_{ik} * a_{jk}) + M * (1 - y_{i,j}), \forall i,j \in J, i \neq j \quad (\text{No overlap}) && (25)\\
& T_j \geq s_j + p_j - d_j (\text{Tardiness definition}) && (26)\\
& T_j \geq 0 (\text{Non-negative tardiness}) && (27)
\end{aligned}
$$

Objective function shows that we minimize the total penalties. The first five constraints are the same as in the task 2a. The last two constraints define tardiness for every job i as $\max\{0, s_i + p_i - d_i\}$.

2.4 Numerical answers

2.4.1 Task 2a

For task 2 a the optimal objective function for instance 1 is:

Minimal completion time is 187.0
Job 1 is starting at 148.0 on machine 2
Job 2 is starting at 124.0 on machine 1
Job 3 is starting at 41.0 on machine 2
Job 4 is starting at 151.0 on machine 2
Job 5 is starting at 130.0 on machine 2
Job 6 is starting at 11.0 on machine 1
Job 7 is starting at 38.0 on machine 2
Job 8 is starting at 36.0 on machine 1
Job 9 is starting at 152.0 on machine 1
Job 10 is starting at 170.0 on machine 1

For instance 2 is:

Minimal completion time is 314.0
Job 1 is starting at 93.0 on machine 6
Job 2 is starting at 185.0 on machine 6
Job 3 is starting at 17.0 on machine 2
Job 4 is starting at 17.0 on machine 10
Job 5 is starting at 59.0 on machine 9
Job 6 is starting at 93.0 on machine 2
Job 7 is starting at 223.0 on machine 7
Job 8 is starting at 2.0 on machine 4

Job 9 is starting at 93.0 on machine 9
Job 10 is starting at 58.0 on machine 3
Job 11 is starting at 194.0 on machine 4
Job 12 is starting at 212.0 on machine 4
Job 13 is starting at 46.0 on machine 8
Job 14 is starting at 196.0 on machine 10
Job 15 is starting at 223.0 on machine 9
Job 16 is starting at 189.0 on machine 6
Job 17 is starting at 122.0 on machine 3
Job 18 is starting at 58.0 on machine 8
Job 19 is starting at 44.0 on machine 5
Job 20 is starting at 223.0 on machine 1
Job 21 is starting at 64.0 on machine 7
Job 22 is starting at 17.0 on machine 1
Job 23 is starting at 223.0 on machine 5
Job 24 is starting at 205.0 on machine 3
Job 25 is starting at 223.0 on machine 3
Job 26 is starting at 131.0 on machine 2
Job 27 is starting at 189.0 on machine 8
Job 28 is starting at 222.0 on machine 8
Job 29 is starting at 170.0 on machine 5
Job 30 is starting at 250.0 on machine 5
Job 31 is starting at 234.0 on machine 2
Job 32 is starting at 138.0 on machine 7
Job 33 is starting at 116.0 on machine 10
Job 34 is starting at 93.0 on machine 1
Job 35 is starting at 102.0 on machine 1
Job 36 is starting at 250.0 on machine 6
Job 37 is starting at 44.0 on machine 7
Job 38 is starting at 57.0 on machine 7
Job 39 is starting at 250.0 on machine 2
Job 40 is starting at 250.0 on machine 10

2.4.2 Task 2b

For task 2b on instance 1:
Job 1 started 55.0 on machine 2 with tardiness 2
Job 2 started 57.0 on machine 1 with tardiness 1
Job 3 started 102.0 on machine 1 with tardiness 1
Job 4 started 58.0 on machine 2 with tardiness 2
Job 5 started 39.0 on machine 1 with tardiness 1
Job 6 started 11.0 on machine 2 with tardiness 2
Job 7 started 35.0 on machine 1 with tardiness 1
Job 8 started 94.0 on machine 2 with tardiness 2
Job 9 started 37.0 on machine 2 with tardiness 2
Job 10 started 85.0 on machine 1 with tardiness 1
Total penalty is 1580.0

The optimal objective function for instance 2 is:
Job 1 started 79.0 on machine 1 with tardiness 1
Job 2 started 152.0 on machine 1 with tardiness 1
Job 3 started 6.0 on machine 3 with tardiness 3
Job 4 started 17.0 on machine 2 with tardiness 2
Job 5 started 64.0 on machine 5 with tardiness 5
Job 6 started 84.0 on machine 5 with tardiness 5
Job 7 started 212.0 on machine 7 with tardiness 7
Job 8 started 2.0 on machine 4 with tardiness 4
Job 9 started 93.0 on machine 2 with tardiness 2
Job 10 started 58.0 on machine 9 with tardiness 9
Job 11 started 194.0 on machine 2 with tardiness 2
Job 12 started 168.0 on machine 1 with tardiness 1
Job 13 started 41.0 on machine 5 with tardiness 5
Job 14 started 199.0 on machine 5 with tardiness 5
Job 15 started 223.0 on machine 8 with tardiness 8
Job 16 started 158.0 on machine 9 with tardiness 9
Job 17 started 125.0 on machine 9 with tardiness 9
Job 18 started 58.0 on machine 3 with tardiness 3
Job 19 started 19.0 on machine 7 with tardiness 7
Job 20 started 125.0 on machine 4 with tardiness 4
Job 21 started 64.0 on machine 6 with tardiness 6
Job 22 started 16.0 on machine 1 with tardiness 1
Job 23 started 222.0 on machine 6 with tardiness 6
Job 24 started 204.0 on machine 9 with tardiness 9
Job 25 started 168.0 on machine 3 with tardiness 3
Job 26 started 131.0 on machine 10 with tardiness 10
Job 27 started 189.0 on machine 6 with tardiness 6
Job 28 started 180.0 on machine 10 with tardiness 10
Job 29 started 145.0 on machine 8 with tardiness 8
Job 30 started 239.0 on machine 4 with tardiness 4
Job 31 started 239.0 on machine 3 with tardiness 3
Job 32 started 79.0 on machine 10 with tardiness 10
Job 33 started 116.0 on machine 5 with tardiness 5
Job 34 started 93.0 on machine 8 with tardiness 8
Job 35 started 85.0 on machine 7 with tardiness 7
Job 36 started 26.0 on machine 8 with tardiness 8
Job 37 started 38.0 on machine 10 with tardiness 10
Job 38 started 36.0 on machine 3 with tardiness 3
Job 39 started 201.0 on machine 8 with tardiness 8
Job 40 started 250.0 on machine 2 with tardiness 2
Total penalty is 4195.0

2.5 Disclosures

3 Problem 3. Car Sharing and Assignment Across Zones

3.1 Problem Description

This problem models a car sharing scheduling system. This is a car sharing scheduling problem. The company wants to serve as many customer requests as possible, minimizing total penalties. And we are given:

- Start time and duration (in minutes)
- Preferred zone (where the request should be served from)
- Eligible zones or vehicles that can serve it
- Penalty 1 if the request is not served at all
- Penalty 2 if the request is served from a neighboring zone (not the preferred one)

So, car-sharing company trying to efficiently manage customer ride requests. Each customer specifies:

- their preferred zone;
- when they need it (start time and duration);
- which specific cars they're willing to use.

The company wants to serve as many requests as possible avoiding next penalties:

- Penalty 1: charged if a request isn't fulfilled at all;
- Penalty 2: charged if the car comes from a nearby zone instead of the preferred one.

Usually, Penalty 1 is greater than Penalty 2.
Rules to follow:

- a car can't be in two places at once (no overlapping bookings);
- only use cars the customer approved;
- prioritize using cars from the customer's preferred zone to avoid the smaller penalty.

So, we must match cars to customers while maximizing served requests, avoid double-booking cars, and keep penalties as low as possible by using preferred zones whenever feasible.

3.2 Model

3.2.1 Parameters

The following parameters are given:

- R — set of customer requests;
- V — set of available vehicles;
- Z — set of zones;
- s_r — start time of request $r \in R$;
- d_r — duration (in minutes) of request $r \in R$;
- z_r^{pref} — preferred zone for request r;
- $Z_r^{\text{eligible}} \subseteq Z$ — eligible zones for serving request r;
- $V_r^{\text{eligible}} \subseteq V$ — eligible vehicles for request r;
- $p_r^{(1)}$ — penalty if request r is not served at all;
- $p_r^{(2)}$ — penalty if request r is served from a non-preferred zone.

3.2.2 Decision Variables and model

We need next decision variables:

- $x_{r,v,z} \in \{0,1\}$ — equals 1 if request r is served by vehicle v from zone z;
- $u_r \in \{0,1\}$ — equals 1 if request r is not served.

Mathematical description is following:

$$\min \quad \sum_{r \in R} \left(p_r^{(1)} \cdot u_r + \sum_{v \in V_r^{\text{eligible}}} \sum_{z \in Z_r^{\text{eligible}}} x_{r,v,z} \cdot \delta_{r,z} \cdot p_r^{(2)} \right) \text{ (Objective function)} \tag{28}$$

$$\text{s.t.} \quad u_r + \sum_{v \in V_r^{\text{eligible}}} \sum_{z \in Z_r^{\text{eligible}}} x_{r,v,z} = 1 \quad \forall r \in R \text{ (Request)} \tag{29}$$

$$x_{r_1,v,z_1} + x_{r_2,v,z_2} \leq 1 \quad \forall z_1, z_2 \in Z \text{ (No overlapping requests)} \tag{30}$$

$$x_{r,v,z} = 0 \quad \text{if } z \notin Z_r^{\text{eligible}} \text{ or } v \notin V_r^{\text{eligible}} \text{ (Eligibility constraint)} \tag{31}$$

where $\delta_{r,z} = 1$ if $z \neq z_r^{\text{pref}}$, otherwise 0.

Objective Function describes minimization of the penalties.

Constraints are the following

1. **Each request is either served or penalized for not being served**
2. **No vehicle may serve overlapping requests**
3. **Only eligible zones and vehicles can serve the request**

3.3 Numerical answers

For task 3 (Toy):
Optimal Assignment: req0: Served by car2,car3,car0,car1,car4 from z0
req1: Served by car2,car3,car1,car4,car5 from z4
req2: Served by car3 from z4
req3: Served by car1 from z0
req4: Served by car2,car3,car4,car5 from z2
req5: Served by car2,car3,car0,car1,car4 from z0
req6: Served by car2 from z3
req7: Served by car3,car0,car1 from z4
req8: Served by car3,car0 from z1
req9: Served by car3 from z4

Total Penalty: 0

For task 3 (Instance 5):
Optimal Assignment: req0: Served by car4,car3,car6,car2,car0,car7,car11,car8,car9,car10,car5 from z27
req1: Served by car4,car3,car6,car2,car0,car7,car11,car8,car9,car10,car5 from z32
req2: Served by car4,car3,car6,car2,car0,car7,car11,car8,car9,car10,car5 from z18
req3: Served by car12 from z34
req4: Served by car4,car3,car6,car2,car0,car7,car11,car8,car9,car10,car5 from z33
req5: Served by car4,car3,car6,car2,car0,car7,car11,car8,car9,car10,car5 from z34

Total Penalty: 0