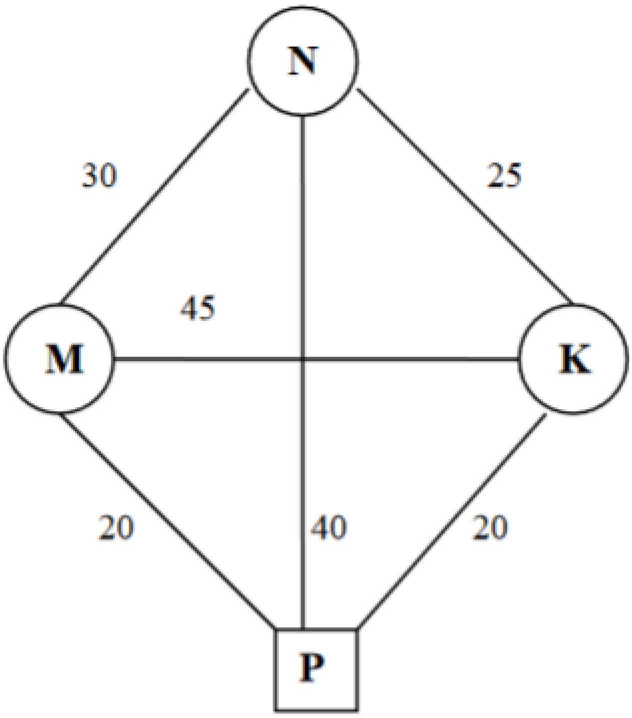
Problem 1. Periodic Vehicle Routing Problem

Consider three customers M, N and K supplied several times during the planning horizon of 3 days with the vehicle capacity of 10 units from supplier P. Customers' total 3-days demands and visit frequencies are given in the table below.

Customer	M	N	K
Total demand	6	9	3
Visit frequency	3	2	2

A road network with travel costs shown on edges is depicted in figure below.



Solve the Periodic Vehicle Routing Problem above using a set-covering model with pre-generation of routes for two cases:

a) When all possible visit schedules for customers are valid;

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $min \sum_{(i,j)\in A} \sum_{k\in K} \sum_{t=1}^{T} c_{ij} x_{ijkt}$	Total_Cost
St	
2) $\sum_{s\in S_i} y_{is} = 1 \ \ \forall\, i \in I$	One_Schedule {i in Customers}
3) $\sum_{r\in R} a_{ir} x_{rt} = \sum_{s\in S_i} b_{ts} y_{is} \ \ \forall\, i \in I, t = 1,..,T$	Day_Schedule {i in Customers, t in 1..T}
4) $x_{rt} \in \{0,1\} \ \ \forall\, r \in R, t = 1,..,T$	x {r in Routes, t in 1..T}
5) $y_{is} \in \{0,1\} \ \ \forall\, i \in I, s \in S_i$	y {i in Customers, s in 1..S}

Notation:

Sets:

R – set of all feasible daily routes	Routes
R_i – subset of routes in set R that contain customer i	r

Parameters:

a_{ir} – binary «coverage» parameter equal to 1 if customer i is visited by route r	a {i in Customers, r in Routes}
c_r – cost of route r	cost {r in Routes}

Variables:

x_{rt} – binary variable equal to 1 if route r is performed on day t	x {r in Routes, t in 1..T}
y_{is} – binary variable equal to 1 if customer i is visited on schedule s	y {i in Customers, s in 1..S}

Description

Objective function (1) minimizes the total cost of all routes during the planning horizon. Constraints (2) ensure that exactly one feasible schedule is chosen for each customer. Constraints (3) guarantee that each customer i is visited only on those days that are specified by the selected visit schedule. Constraints (4) and (5) ensure that variables x and y are binary.

Problem size

The resulting model has following dimensions:

2 variables
4 constraints

2) AMPL code

```
File task4.dat:
param T := 3;
set Customers := M N K;
param S := 4;
#set Schedules := A B C D;

param b: 1 2 3 4:=
      1    1 0 1 1
      2    1 1 0 1
      3  0 1 1 1;

set Schedule[M] := 4;
set Schedule[N] := 1 2 3;
set Schedule[K] := 1 2 3;

set Routes := a b c d e f g;

param a: a b c d e f g:=
      M  1 0 0 1 0 1 1
      N  0 1 0 1 1 0 1
      K    0 0 1 0 1 1 1;

param cost := a 40, b 80, c 40, d 90, e 85, f 85, g 95;

File task4.mod:
```

```
param T >=0;
set Customers;
param S >=0;
#set Schedules {s in 1..S};
param b {t in 1..T, s in 1..S} binary;
set Routes;
param cost {r in Routes};
param a {i in Customers, r in Routes} binary;
set Schedule {i in Customers} within {1..S};
var x {r in Routes, t in 1..T} binary;
var y {i in Customers, s in 1..S} binary;

minimize Total_Cost:
     sum {t in 1..T, r in Routes} cost[r]*x[r,t];
subject to One_Schedule {i in Customers}:
     sum{s in Schedule[i]} y[i,s] = 1;
subject to Day_Schedule {i in Customers, t in 1..T}:
     sum {r in Routes} a[i,r]*x[r,t] = sum {s in Schedule[i]}
b[t,s]*y[i,s];
```

File task4.run

```
option solver cplex;
#option cplex_options 'sensitivity';
model task4.mod;
data task4.dat;
solve;
#option omit_zero_rows 1;
#option show_stats 1;
display Total_Cost > task4.sol;
display x > task4.sol;
display y > task4.sol;
```

File task4.sol

```
Total_Cost = 230

x [*,*]
:   1   2   3    :=
a   0   1   0
b   0   0   0
c   0   0   0
d   0   0   0
e   0   0   0
f   0   0   0
g   1   0   1
;
```

```
y :=
K 1   0
K 2   0
K 3   1
K 4   0
M 1   0
M 2   0
M 3   0
M 4   1
N 1   0
N 2   0
N 3   1
N 4   0
;
```

3) Solution

To minimize the total cost, the following schedule should be implemented: customer M is visited every day, customers N and K – on the first and the third day. The chosen routes are:

- On day 1 – P-M-N-K-P;
- On day 2 – P-M-P;
- On day 3 – P-M-N-K-P.

Total cost in this case will be 230.

b) When customers N and K cannot be visited in two consecutive days.

2) AMPL code

```
File task4.dat:
set Schedule[M] := 4;
set Schedule[N] := 3;
set Schedule[K] := 3;

File task2a.sol
Total_Cost = 230

x [*,*]
:   1   2   3    :=
a   0   1   0
b   0   0   0
c   0   0   0
d   0   0   0
e   0   0   0
f   0   0   0
```

```
g   1   0   1
;

y :=
K 1   0
K 2   0
K 3   1
K 4   0
M 1   0
M 2   0
M 3   0
M 4   1
N 1   0
N 2   0
N 3   1
N 4   0
;
```

3) Solution

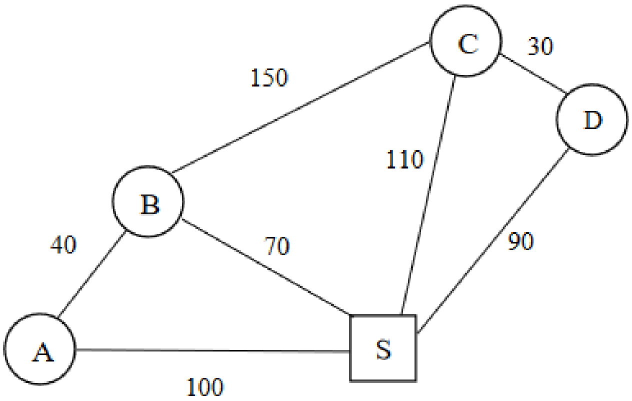
The solution from point a) already included visiting customers N and K every other day, so nothing changes here. To minimize the total cost, the following schedule should be implemented: customer M is visited every day, customers N and K – on the first and the third day. The chosen routes are:

- On day 1 – P-M-N-K-P;
- On day 2 – P-M-P;
- On day 3 – P-M-N-K-P.

Total cost is still 230.

Problem 2. Inventory Routing Problem

Consider a network with a single supplier S and four customers A, B, C and D depicted on the graph below with travel costs shown on edges, and customers' daily consumption rates and storage capacities in units given in the following table.



Customer i	Capacity C_i	Consumption q_i
A	180	70
B	100	100
C	150	150
D	300	50

The planning horizon is 4 days and there are unlimited number of vehicles with capacity Q = 250 units. Assume that we consider the direct routes to a customer and the routes that serve adjacent customers, and it's assumed that each route can be performed at most once per day. Assume for the beginning that there are no initial inventories at customers, deliveries take place before consumption, and inventories are measured at the end of the day.

Find a cheapest distribution policy (show routes for each day and how much will be delivered/stored at customers) for the Inventory Routing Problem for each of the following cases:

a) when only travel costs are minimized;

2) AMPL code

```
File task5a.dat:
param K:=4;
param T:=4;
param veh_cap:= 250;
param consumption:= 1 70 2 100 3 150 4 50;
param inv_cap:= 1 180 2 100 3 150 4 300;
set Routes := 1 2 3 4 12 23 34;
param cost:= 1 200 2 140 3 220 4 180 12 210 23 330 34 230;

set Route[1]:= 1 12;
set Route[2]:= 2 12 23;
set Route[3]:= 3 23 34;
set Route[4]:= 4 34;

File task5a.mod:
param K;
param T;
set Customers:= {i in 1..K};
set Routes;

param veh_cap;
param consumption {Customers};
param inv_cap {Customers};
param cost {Routes};

set Route {i in Customers} within Routes;
```

```
var x {k in Routes, t in 1..T}  binary;
var amount {i in Customers, k in Routes, t in 1..T} >= 0;
var inv {i in Customers, t in 0..T};

minimize Total_Cost:
     1 / T * (sum{t in 1..T} sum {k in Routes} cost[k] * x[k,t]);

subject to Vehicle_Capacity{t in 1..T, i in Customers, k in
Route[i]}:
     sum {p in Customers} amount[p,k,t] <=  veh_cap * x[k,t];

subject to Inventory_Level{i in Customers,  t in 1..T}:
     inv[i,t] = inv[i,t-1] + sum{k in Route[i]} amount[i,k,t]
- consumption[i];

subject to Inventory_Capacity {i in Customers, t in 1..T}:
     inv[i,t] + consumption[i] <= inv_cap[i];

subject to Initial_Inventory {i in Customers}:
     inv[i,0]  = 0;

subject to No_Stockouts {i in Customers, t in 0..T}:
     inv[i,t]  >= 0;
```

File task5a.run:

```
option solver cplex;
model task5.mod;
data task5.dat;
option show_stats 1;
solve;
display Total_Cost > task5.sol;
display x > task5.sol;
display amount > task5.sol;
display inv > task5.sol;
```

File task5a.sol:

```
Total_Cost = 385

x [*,*]
:    1   2   3   4    :=
1    0   0   0   0
2    0   0   0   0
3    0   0   0   0
4    0   0   0   0
12   1   0   1   0
```

```
23   0   1   0   1
34   1   0   1   0
;

amount [1,*,*]
:     1    2    3    4    :=
1      0   0     0   0
2      0   0     0   0
3      0   0     0   0
4      0   0     0   0
12   140   0   140   0
23     0   0     0   0
34     0   0     0   0

 [2,*,*]
:     1     2     3     4     :=
1      0     0     0     0
2      0     0     0     0
3      0     0     0     0
4      0     0     0     0
12   100     0   100     0
23     0   100     0   100
34     0     0     0     0

 [3,*,*]
:     1     2     3     4     :=
1      0     0     0     0
2      0     0     0     0
3      0     0     0     0
4      0     0     0     0
12     0     0     0     0
23     0   150     0   150
34   150     0   150     0

 [4,*,*]
:     1    2    3    4    :=
1      0   0     0   0
2      0   0     0   0
3      0   0     0   0
4      0   0     0   0
12     0   0     0   0
23     0   0     0   0
34   100   0   100   0
;
```

```
inv :=
1 0    0
1 1   70
1 2    0
1 3   70
1 4    0
2 0    0
2 1    0
2 2    0
2 3    0
2 4    0
3 0    0
3 1    0
3 2    0
3 3    0
3 4    0
4 0    0
4 1   50
4 2    0
4 3   50
4 4    0
;
```

3) Solution

Routes for each day are the following:

- Day 1: S-A-B-S, S-C-D-S;
- Day 2: S-B-C-S;
- Day 3: S-A-B-S, S-C-D-S;
- Day 4: S-B-C-S.

Average daily cost in this case will be 385.

b) when both travel costs and inventory holding costs at customers are minimized (unit holding costs are 1);

2) AMPL code

```
File task5b.dat:
param hold_cost:= 1;

File task5b.mod:
param hold_cost;
minimize Total_Cost:
     1 / T * (sum{t in 1..T} sum {k in Routes} cost[k] * x[k,t]
+ sum{t in 1..T} sum{i in Customers} inv[i,t] * hold_cost);

file task5b.sol:
```

```
Total_Cost = 440

x [*,*]
:    1   2   3   4    :=
1    0   0   0   0
2    0   0   0   0
3    0   0   0   0
4    0   0   0   0
12   1   1   1   1
23   0   0   0   0
34   1   1   1   1
;

amount [1,*,*]
:    1    2    3    4     :=
1     0    0    0    0
2     0    0    0    0
3     0    0    0    0
4     0    0    0    0
12   70   70   70   70
23    0    0    0    0
34    0    0    0    0

 [2,*,*]
:     1     2     3     4     :=
1      0     0     0     0
2      0     0     0     0
3      0     0     0     0
4      0     0     0     0
12   100   100   100   100
23     0     0     0     0
34     0     0     0     0

 [3,*,*]
:     1     2     3     4     :=
1      0     0     0     0
2      0     0     0     0
3      0     0     0     0
4      0     0     0     0
12     0     0     0     0
23     0     0     0     0
34   150   150   150   150

 [4,*,*]
:    1    2    3    4     :=
```

```
1      0    0    0    0
2      0    0    0    0
3      0    0    0    0
4      0    0    0    0
12     0    0    0    0
23     0    0    0    0
34    50   50   50   50
;

inv :=
1 0   0
1 1   0
1 2   0
1 3   0
1 4   0
2 0   0
2 1   0
2 2   0
2 3   0
2 4   0
3 0   0
3 1   0
3 2   0
3 3   0
3 4   0
4 0   0
4 1   0
4 2   0
4 3   0
4 4   0
;
```

3) Solution

In this case the routes for each day will be the same: S-A-B-S and S-C-D-S.
Average daily cost of this solution is 440.

c) when travel costs and inventory holding costs at customers are minimized and initial inventory levels at customers are found by optimization (unit holding costs are 0.5).

2) AMPL code

```
File task5c.dat:
param hold_cost:= 0.5;

File task5c.mod:
```

```
var start_inv {i in Customers} >=0;

minimize Total_Cost:
     1 / T * (sum{t in 1..T} sum {k in Routes} cost[k] * x[k,t]
     + sum{t in 1..T} sum{i in Customers} inv[i,t] * hold_cost
     + sum{i in Customers} start_inv[i] * hold_cost);

File task5c.sol:
Total_Cost = 415

start_inv [*] :=
1  0
2  0
3  0
4  0
;

x [*,*]
:    1   2   3   4    :=
1    0   0   0   0
2    0   0   0   0
3    0   0   0   0
4    0   0   0   0
12   1   0   1   0
23   0   1   0   1
34   1   0   1   0
;

amount [1,*,*]
:     1    2    3    4    :=
1      0   0     0   0
2      0   0     0   0
3      0   0     0   0
4      0   0     0   0
12   140   0   140   0
23     0   0     0   0
34     0   0     0   0

 [2,*,*]
:     1     2     3     4     :=
1      0     0     0     0
2      0     0     0     0
3      0     0     0     0
4      0     0     0     0
12   100     0   100     0
```

```
23      0   100      0   100
34      0     0      0     0

 [3,*,*]
:      1     2     3     4     :=
1      0     0     0     0
2      0     0     0     0
3      0     0     0     0
4      0     0     0     0
12     0     0     0     0
23     0   150     0   150
34   150     0   150     0

 [4,*,*]
:      1    2    3    4    :=
1      0    0      0   0
2      0    0      0   0
3      0    0      0   0
4      0    0      0   0
12     0    0      0   0
23     0    0      0   0
34   100    0    100   0
;

inv :=
1 0    0
1 1   70
1 2    0
1 3   70
1 4    0
2 0    0
2 1    0
2 2    0
2 3    0
2 4    0
3 0    0
3 1    0
3 2    0
3 3    0
3 4    0
4 0    0
4 1   50
4 2    0
4 3   50
4 4    0
```

;

3) Solution

Routes for each day are the same as in a):

- Day 1: S-A-B-S, S-C-D-S;
- Day 2: S-B-C-S;
- Day 3: S-A-B-S, S-C-D-S;
- Day 4: S-B-C-S.

Optimal inventory level is 0.

Average daily cost of this solution is 415.