

```
In [1]: import numpy as np
import sys
from docplex.cp.model import *
```

```
In [2]: NumberCustomers = 10
Capacity = 200
Customers = [i for i in range(1, NumberCustomers + 1)]
CustomersDepot = [0] + Customers + [NumberCustomers + 1] ##index 0 and 11 are depot
Vehicles = [i for i in range(1, 4)] #NV
ReadyTime=[0,74,151,116,42,65,42,90,167,95,180,0]
DueTime = [240,104,181,146,72,95,72,120,197,125,210,240]
ser=[0,10,10,10,10,10,10,10,10,10,0] # Service Time
PDemand=[0,10,40,10,20,23,16,23,3,25,26,0] # Pick up
DDemand=[0,10,10,30,19,14,9,13,13,23,3,0] # Delivery
```

```
In [3]: XCOORD=[40,88,42,72,10,65,27,12,49,57,31,40]
YCOORD=[50,30,5,35,20,55,43,24,42,48,67,50]
A = [(i, j) for i in CustomersDepot for j in CustomersDepot ]
#print(A)
TravelDistance = {(i, j):np.hypot(XCOORD[i]-XCOORD[j], YCOORD[i]-YCOORD[j]) for i, j in A}
#TravelDistance[1,2]
```

```
TravelDistance
(10, 5): 36.05551275463989,
(10, 6): 24.331050121192877,
(10, 7): 47.01063709417264,
(10, 8): 30.805843601498726,
(10, 9): 32.202484376209235,
(10, 10): 0.0,
(10, 11): 19.235384061671343,
(11, 0): 0.0,
(11, 1): 52.0,
(11, 2): 45.044422518220834,
(11, 3): 35.34119409414458,
(11, 4): 42.42640687119285,
(11, 5): 25.495097567963924,
(11, 6): 14.7648230602334,
(11, 7): 38.2099463490856,
(11, 8): 12.041594578792296,
(11, 9): 17.11724276862369,
(11, 10): 19.235384061671343,
(11, 11): 0.0}
```

```
In [4]: #d=[[int(TravelDistance[i,j]) for i in CustomersDepot] for j in CustomersDepot ]
#d
```

```
In [5]: mdl = CpoModel()
```

```
In [6]: x_i_k = {}
for t in CustomersDepot:
    for v in Vehicles:
        iv=mdl.interval_var(start=(ReadyTime[t], INTERVAL_MAX), end=(0, DueTime[t]), size=ser[t])
        iv.set_optional()
        x_i_k[(t,v)]=iv
        print(x_i_k[(t,v)])
#x_i_k
```

```
intervalVar(optional, end=0..240, size=0)
intervalVar(optional, end=0..240, size=0)
intervalVar(optional, end=0..240, size=0)
intervalVar(optional, start=74..intervalmax, end=0..104, size=10)
intervalVar(optional, start=74..intervalmax, end=0..104, size=10)
intervalVar(optional, start=74..intervalmax, end=0..104, size=10)
intervalVar(optional, start=151..intervalmax, end=0..181, size=10)
intervalVar(optional, start=151..intervalmax, end=0..181, size=10)
intervalVar(optional, start=151..intervalmax, end=0..181, size=10)
intervalVar(optional, start=116..intervalmax, end=0..146, size=10)
intervalVar(optional, start=116..intervalmax, end=0..146, size=10)
intervalVar(optional, start=116..intervalmax, end=0..146, size=10)
intervalVar(optional, start=42..intervalmax, end=0..72, size=10)
intervalVar(optional, start=42..intervalmax, end=0..72, size=10)
intervalVar(optional, start=42..intervalmax, end=0..72, size=10)
intervalVar(optional, start=65..intervalmax, end=0..95, size=10)
intervalVar(optional, start=65..intervalmax, end=0..95, size=10)
intervalVar(optional, start=65..intervalmax, end=0..95, size=10)
intervalVar(optional, start=42..intervalmax, end=0..72, size=10)
intervalVar(optional, start=42..intervalmax, end=0..72, size=10)
intervalVar(optional, start=42..intervalmax, end=0..72, size=10)
intervalVar(optional, start=90..intervalmax, end=0..120, size=10)
intervalVar(optional, start=90..intervalmax, end=0..120, size=10)
intervalVar(optional, start=90..intervalmax, end=0..120, size=10)
intervalVar(optional, start=167..intervalmax, end=0..197, size=10)
intervalVar(optional, start=167..intervalmax, end=0..197, size=10)
intervalVar(optional, start=167..intervalmax, end=0..197, size=10)
intervalVar(optional, start=95..intervalmax, end=0..125, size=10)
intervalVar(optional, start=95..intervalmax, end=0..125, size=10)
intervalVar(optional, start=95..intervalmax, end=0..125, size=10)
intervalVar(optional, start=180..intervalmax, end=0..210, size=10)
intervalVar(optional, start=180..intervalmax, end=0..210, size=10)
intervalVar(optional, start=180..intervalmax, end=0..210, size=10)
intervalVar(optional, end=0..240, size=0)
intervalVar(optional, end=0..240, size=0)
intervalVar(optional, end=0..240, size=0)
```

```
In [7]: y_i = {}
for t in CustomersDepot:
    y_i[t]= mdl.interval_var(start=(ReadyTime[t], INTERVAL_MAX), end=(0, DueTime[t]), size=ser[t])
    print(y_i[t])
#y_i
```

```
intervalVar(end=0..240, size=0)
intervalVar(start=74..intervalmax, end=0..104, size=10)
intervalVar(start=151..intervalmax, end=0..181, size=10)
intervalVar(start=116..intervalmax, end=0..146, size=10)
intervalVar(start=42..intervalmax, end=0..72, size=10)
intervalVar(start=65..intervalmax, end=0..95, size=10)
intervalVar(start=42..intervalmax, end=0..72, size=10)
intervalVar(start=90..intervalmax, end=0..120, size=10)
intervalVar(start=167..intervalmax, end=0..197, size=10)
intervalVar(start=95..intervalmax, end=0..125, size=10)
intervalVar(start=180..intervalmax, end=0..210, size=10)
intervalVar(end=0..240, size=0)
```

```
In [8]: Q={}
for v in Vehicles:
    Q[v]=mdl.sequence_var([x_i_k[(t, v)] for t in CustomersDepot], types=([t for t in CustomersDepot]))
    print(Q[v])
```

```
sequenceVar([Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
sequenceVar([Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
sequenceVar([Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous, Anonymous], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
```

```
In [9]: ff={}
for v in Vehicles:
    ff[v]=sum mdl.step_at_start(x_i_k[t,v], PDemand[t]) for t in Customers )
    - sum mdl.step_at_start(x_i_k[t,v], DDemand[t]) for t in Customers)

    print(ff[v])

#for v in Vehicles:
#    sum mdl.add mdl.presence_of(x_i_k[(t,v)]) * PDemand[t] for t in Customers )
#    + (mdl.step_at_start(x_i_k[t,v], PDemand[t]) for t in Customers )
#    - (mdl.step_at_start(x_i_k[t,v], DDemand[t]) for t in Customers)
```

```
sum([stepAtStart(intervalVar(optional, start=74..intervalmax, end=0..104, size=10), 10), stepAtStart(intervalVar(optional, start=151..intervalmax, end=0..181, size=10), 40), stepAtStart(intervalVar(optional, start=116..intervalmax, end=0..146, size=10), 10), stepAtStart(intervalVar(optional, start=42..intervalmax, end=0..72, size=10), 20), stepAtStart(intervalVar(optional, start=65..intervalmax, end=0..95, size=10), 23), stepAtStart(intervalVar(optional, start=42..intervalmax, end=0..72, size=10), 16), stepAtStart(intervalVar(optional, start=90..intervalmax, end=0..120, size=10), 23), stepAtStart(intervalVar(optional, start=167..intervalmax, end=0..197, size=10), 3), stepAtStart(intervalVar(optional, start=95..intervalmax, end=0..125, size=10), 25), stepAtStart(intervalVar(optional, start=180..intervalmax, end=0..210, size=10), 26)])
sum([stepAtStart(intervalVar(optional, start=74..intervalmax, end=0..104, size=10), 10), stepAtStart(intervalVar(optional, start=151..intervalmax, end=0..181, size=10), 40), stepAtStart(intervalVar(optional, start=116..intervalmax, end=0..146, size=10), 10), stepAtStart(intervalVar(optional, start=42..intervalmax, end=0..72, size=10), 20), stepAtStart(intervalVar(optional, start=65..intervalmax, end=0..95, size=10), 23), stepAtStart(intervalVar(optional, start=42..intervalmax, end=0..72, size=10), 16), stepAtStart(intervalVar(optional, start=90..intervalmax, end=0..120, size=10), 23), stepAtStart(intervalVar(optional, start=167..intervalmax, end=0..197, size=10), 3), stepAtStart(intervalVar(optional, start=95..intervalmax, end=0..125, size=10), 25), stepAtStart(intervalVar(optional, start=180..intervalmax, end=0..210, size=10), 26)])
sum([stepAtStart(intervalVar(optional, start=74..intervalmax, end=0..104, size=10), 10), stepAtStart(intervalVar(optional, start=151..intervalmax, end=0..181, size=10), 40), stepAtStart(intervalVar(optional, start=116..intervalmax, end=0..146, size=10), 10), stepAtStart(intervalVar(optional, start=42..intervalmax, end=0..72, size=10), 20), stepAtStart(intervalVar(optional, start=65..intervalmax, end=0..95, size=10), 23), stepAtStart(intervalVar(optional, start=42..intervalmax, end=0..72, size=10), 16), stepAtStart(intervalVar(optional, start=90..intervalmax, end=0..120, size=10), 23), stepAtStart(intervalVar(optional, start=167..intervalmax, end=0..197, size=10), 3), stepAtStart(intervalVar(optional, start=95..intervalmax, end=0..125, size=10), 25), stepAtStart(intervalVar(optional, start=180..intervalmax, end=0..210, size=10), 26)])
```

```
In [10]: for v in Vehicles:
        mdl.add(ff[v] <= Capacity)
```

```
In [11]: for t in Customers:
        mdl.add(mdl.alternative(y_i[t], [x_i_k[(t,v)] for v in Vehicles]))
```

```
In [12]: #transitionTimes = transition_matrix([int(TravelDistance[i,j]) for i in CustomersDepot] for j in CustomersDepot)
#transitionTimes
#d=[[int(TravelDistance[i,j]) for i in Customers] for j in Customers ]
#d
```

```
In [13]: for v in Vehicles:
        mdl.add(mdl.no_overlap(Q[v]))
```

```
In [14]: for v in Vehicles:
        mdl.add(mdl.presence_of(x_i_k[(0,v)])==1)
        mdl.add(mdl.presence_of(x_i_k[(11,v)])==1)
        mdl.add(mdl.first(Q[v], x_i_k[(0,v)]))
        mdl.add(mdl.last(Q[v], x_i_k[(11,v)]))
```

```
In [15]: obj=sum((mdl.element([TravelDistance[t, t_dest] for t_dest in CustomersDepot],
        mdl.type_of_next(Q[v], x_i_k[(t, v)], t, t)) for v in Vehicles for t in CustomersDepot))
obj
```

```
Out[15]: <docplex.cp.expression.CpoFunctionCall at 0x7fcd3c8479b0>
```

```
In [16]: mdl.add(minimize(obj))
```

In [17]: `## Solve the model`

```
mdl.solve(TimeLimit=5,execfile='/Applications/CPLEX_Studio_Beta211/coptimizer/bin/x86-64_osx/coptimize
```

```
! Time = 4.70s, Average fail depth = 14, memory usage = 4.1 MB
! Current bound is 35.34119 (gap is 88.57%)
!
!      Best Branches  Non-fixed   W      Branch decision
!      309.2597      46000       24     1   F      74 >= startOf(_ITV_15)
!      309.2597      47000       17     1   F           presenceOf(_ITV_4)
!      309.2597      48000       19     1   F      50 <= startOf(_ITV_14)
!      309.2597      63000        1     2           90 = startOf(_ITV_27)
!
! -----
! Search terminated by limit, 11 solutions found.
! Best objective      : 309.2597 (gap is 88.57%)
! Best bound          : 35.34119
!
! -----
! Number of branches   : 248312
! Number of fails      : 121352
! Total memory usage   : 4.3 MB (4.2 MB CP Optimizer + 0.1 MB Concert)
! Time spent in solve  : 5.04s (4.93s engine + 0.12s extraction)
! Search speed (br. / s) : 50367.5
! -----
```

Out [17]: `<docplex.cp.solution.CpoSolveResult at 0x7fcd3c8515f0>`

In [ ]:

In [ ]: