

# 1 GRASP

- Input:  $n \in N$ ,  $h \in H$ , minHours, maxHours, maxConsec, maxPresence
- Output:  $W_{n,h}, \text{cost}$
- Procedure Construct
  - (1) Initialize  $W_{n,h} = 0, \forall n \in N, \forall h \in H$
  - (2) for n in N
  - (3)   ComputeGreedyCost(H)
  - (4)   order1=sort(H,'gc',asc)
  - (5)    $\min_{gc}$ =the gc of of first(order1),  $\max_{gc}$ =the gc of the last(order1)
  - (6)    $\text{RCL} \leq \min_{gc} + \alpha(\max_{gc} - \min_{gc})$
  - (7)   Randomly select an hour (h) from RCL, set  $W_{n,h}=1$ , remove(h)
  - (8)   if complete(All  $gc_h \geq 0$ )   then **break**
  - (9)   while k≤max
  - (10)     get the  $h_{start}$  &  $h_{end}$  of the working hours
  - (11)     if Sum( $w_{n,h}$ ) ≥ min &  $h_{end} - h_{start} + 1 == \text{maxPresence}$    then **break**
  - (12)     order2=sort(H[ $h_{start}-2, h_{end} + 2$ ],'gc',asc)
  - (13)     if satisfy maxConsec
  - (14)       h'=first(order2), set  $W_{n,h'}=1$
  - (15)     remove(h')
  - (16) cost = getNumberOfWorkingNurses
  - (17) return  $W_{n,h}$ , cost
 End Construct
- Procedure Local Search
  - (1)  $W''_{n,h} = W_{n,h}$
  - (2) for n ∈ NumberofWorkingNurse
  - (3)  $W'_{n,h} = W''_{n,h}$
  - (4)   for n' ∈ NumberofWorkingNurse \ n
  - (5)     Set  $W'_{n,h}=0, W'_{n',h}=0, \forall h \in H$
  - (6)     if SumofWorkinghours( $\forall h \in H$ ) ≤ Demand<sub>h</sub>-2   then **continue**
  - (7)     elseif CanNotGenerateFeasibleW'   then **continue**
  - (8)     Generate a Feasible  $W'_{n,h}$
  - (9)      $W''_{n',h} = W'_{n',h}, W''_{n,h} = W'_{n,h}$
  - (10)    cost-=1
  - (11)    **break**

(12)    Update NumberOfWorkingNurse  
(13)  $W_{n,h}=W''_{n,h}$   
(14) return  $W_{n,h}$ , cost  
End Local Search