Liner shipping hub location-routing problem under demand uncertainty

Cihan Butun *[1], Sanja Petrovic[2], and Luc Muyldermans[3]

[1]Loughborough Business School, Loughborough University, Loughborough, LE11 3TU, United Kingdom. Email: c.butun@lboro.ac.uk
[2]Nottingham University Business School, Jubilee Campus, Nottingham NG81BB, United Kingdom, Email: sanja.petrovic@nottingham.ac.uk
[3]Nottingham University Business School, Jubilee Campus, Nottingham NG81BB, United Kingdom, Email: luc.muyldermans@nottingham.ac.uk

Algorithm pseudo-codes

1. Tabu Search

Table 1: Tabu Search parameters

Parameter	**Explanation**
$\mathcal{L}^o$	Set of open hubs in the initial solution
$\mathcal{L}^c$	Set of closed hubs in the initial solution
$\mathcal{U}$	Set of unallocated non-hub nodes in the initial solution
tf	Total flows routed in the network, $tf = \sum_{k=1}^{H}(1-\beta)(o_{k(3)} + d_{k(3)}) + (\beta - \lambda)(o_{k(4)} + d_{k(4)})$
$tcap$	Total cargo handling capacity of hub candidates, $tcap = \sum_{k=1}^{H} cap_k$
I_k	Hub attractiveness score of $k,\ k \in \mathcal{H}$
sd_{ki}	The sea distance between non-hub terminal i and hub candidate terminal k in nautical miles, $k \in \mathcal{H},\ i \in \mathcal{N}$
$volumeFactor$	The multiplier that indicates the minimum total capacity that the located hubs must have in the initial solution, $1 \leq volumeFactor \leq \frac{tcap}{tf}$
$iter$	Current number of consecutive iterations without improvement
$maxIter$	The Tabu Search algorithm terminates when $iter = maxIter$
$openFreq$	The list of hub location frequencies of currently open hubs
$maxFreq$	The maximum value of hub opening frequencies in openFreq
$limFreq$	Diversification is triggered when $maxFreq \geq limFreq$
S	A feasible solution
S_0	The initial feasible solution
S_{best}	The best solution obtained in the Tabu Search algorithm
tnc_s	The network cost of solution S

*Corresponding author

Algorithm 1 Initial solution procedure

$S_0 \leftarrow \emptyset$, $tnc_{S_0} \leftarrow \infty$, $volumeFactor \leftarrow 1$
for all $k \in \mathcal{H}$ **do**
  Calculate I_k
end for
while $volumeFactor \leq tcap/tf$ **do**
  $\mathcal{U} \leftarrow \mathcal{N}$, $\mathcal{L}^o \leftarrow \emptyset$, $\mathcal{L}^c \leftarrow \mathcal{H}$
  Rank $k \in \mathcal{L}^c$ in non-increasing order of I_k
  while $\sum_{k=1}^{|\mathcal{L}^o|} cap_k < volumeFactor \cdot tf$ **do**
    Open a new hub at the top element $k \in \mathcal{L}^c$
    Remove k from $\mathcal{L}^c$ and $\mathcal{U}$
    Add k to $\mathcal{L}^o$
  end while
  while $\mathcal{U} \neq \emptyset$ **do**
    for all $k \in \mathcal{L}^o$ and $i \in \mathcal{U}$ **do**
      if k has enough capacity to accommodate i **then**
        Allocate i to k (subject to sufficient capacity) $\mid sd_{ki} = \min_{\forall k \in \mathcal{L}^o} sd_{ki}$
        Remove i from $\mathcal{U}$
      end if
      if i cannot be allocated to any $k \in \mathcal{L}^o$ **then**
        Discard the solution
        break
      end if
    end for
  end while
  if $\mathcal{U} = \emptyset$ **then**
    Update S with the hub locations and node allocations
    Apply Nearest-Neighbor heuristic to obtain the hub-level network for S
    if $tnc_S < tnc_{S_0}$ **then**
      $S_0 \leftarrow S$, $tnc_{S_0} \leftarrow tnc_S$
    end if
  end if
  $volumeFactor \leftarrow volumeFactor + 0.05$
end while
return S_0

Algorithm 2 Tabu Search algorithm

Run the Initial solution procedure and obtain S_0
$iter \leftarrow 1$, $S_{iter}, S_{best} \leftarrow S_0$, $tnc_{S_{best}} \leftarrow tnc_{S_0}$
while $iter \neq maxIter$ **do**
    Do a neighborhood search and update S_{iter}
    Run local improvement with 2-opt and update S_{iter}
    if $tnc_{S_{iter}} < tnc_{S_{best}}$ **then**
        $S_{best} \leftarrow S_{iter}$, $tnc_{S_{best}} \leftarrow tnc_{S_{iter}}$, $iter \leftarrow 0$
    else
        $iter \leftarrow iter + 1$
    end if
    Update $openFreq$ and Tabu lists
    if $maxFreq \geq limFreq$ **then**
        Run $Diversification$
        $limFreq \leftarrow maxFreq + limFreq$
        Update Tabu lists
    end if
end while
return S_{best}

2. Decomposition Heuristic

Table 2: Decomposition heuristic parameters

Parameter	Explanation
t	The current iteration number
tnc_{DH}	The lowest total network cost obtained by the decomposition heuristic
$SP1$	Subproblem 1, locates hubs and determines non-hub to hub allocations
$SP2$	Subproblem 2, designs the hub-level network and determines cargo routing between hubs.
$SP3$	Subproblem 3, re-optimizes non-hub to hub allocations terminals.
p_t	The number of hubs to be located at iteration t
p_{max}	The current maximum number of hubs that can be located
θ_t^{SP1}	The objective function value of SP1 calculated at iteration t with $p = p_t$
$\theta_{p_{max}}^{SP1}$	The objective function value of SP1 calculated with $p = p_{max}$
θ_t^{SP2}	The objective function value of SP2 calculated at iteration t
θ_t^{SP3}	The objective function value of SP3 calculated at iteration t

Algorithm 3 Decomposition heuristic

$t \leftarrow 1$, $tnc_{DH} \leftarrow \infty$, $p_{max} \leftarrow H$, $p_t \leftarrow 2$
while $p_t < p_{max}$ **do**
    Solve SP1 with $p = p_t$, obtain θ_t^{SP1}
    if $t = 1$ or ($t > 1$ and $\theta_t^{SP1} < \theta_{t-1}^{SP1}$) **then**
        Solve SP2, obtain θ_t^{SP2}
        Solve SP3, obtain θ_t^{SP3}
        if $\theta_t^{SP3} < tnc_{DH}$ **then**
            $tnc_{DH} \leftarrow \theta_t^{SP3}$
            while true **do**
                Solve SP1 with $p = p_{max}$, obtain $\theta_{p_{max}}^{SP1}$
                if $\theta_{p_{max}}^{SP1} \geq tnc_{DH}$ **then**
                    $p_{max} \leftarrow p_{max} - 1$
                else
                    break
                end if
            end while
        end if
    end if
    $t \leftarrow t + 1$, $p_t \leftarrow p_t + 1$
end while
return tnc_{DH}