A Threshold Accepting Algorithm for the Uncapacitated Single Allocation Hub Location Problem with Concave Function

Hung-Jie Wang
Powerchip Technology Corporation, Hsinchu, Taiwan 30078, R.O.C.
Tel: (+886) 3-5791000-2931
Email: hjwang@powerchip.com

Ching-Jung Ting†
Department of Industrial Engineering and Management
Yuan Ze University, Chung-Li, Taiwan 32003, R.O.C.
Tel: (+886) 3-4638800-2526 Fax: (+886) 3-4638907
Email: ietingcj@saturn.yzu.edu.tw

Abstract. Today, the hub-and-spoke (H&S) network is a general application in many transportation networks. In this paper, the uncapacitated single allocation hub location problem (USAHL) with concave costs for the interhub arcs is considered. Most of the previous research assumes the transportation cost between hubs has a constant discount factor. However, larger consolidated flow should have better economic of scales of transportation cost in the real application. We relax the constant discount factor in this research. The USAHL with concave cost is formulated as a mixed integer programming model. Since USAHL is NP-hard, we propose a threshold accepting (TA) algorithm to solve the problem. Our algorithm was tested on two sets of benchmark instances, Civil Aeronautics Board (CAB) and Australia Post (AP). The results and comparison with a lower bound are reported.

Keywords: hub-and-spoke network, hub location, threshold accepting, concave function

1. INTRODUCTION

The hub-and-spoke (H&S) network has been applied in different fields, such as air transportation, telecommunications, and freight delivery industry (Campbell et al., 2002). On these applications, instead of these client nodes exchange flows directly, a subset of client nodes is chosen to act as consolidation points or switching points (hubs). Thus, a bundle of flows from different origins is routed via one or two hubs before being delivered to the same destination. Consequently, a bundled flow on interconnected hubs achieves the larger economies of scale to share transportation costs. Since O’Kelly (1987) introduced the pioneering works, the variant problems are presented with different types of model for locating hubs and assigning non-hub clients to hubs. Several interesting surveys can be found in Campbell (1994), Campbell et al. (2002), and Alumur and Kara (2008) for details.

A well-known problem, called uncapacitated single allocation hub location problem (USAHL), was introduced by O’Kelly (1992), where the number of hubs is endogenous with considering the installation costs, and each non-hub node should assign to only one hub depending on its geography and traffic. Furthermore, there is a discount factor, 0 < α < 1, on each interhub link for lowering the unit transportation cost. The decision of the USAHL is to decide the number and location of hubs, and then allocate the non-hub nodes to the best cluster to minimize the total transportation. Regarding the USAHL, many algorithms had been proposed (Abdinnour-Helm and Venkataramanan, 1998; Abdinnour-Helm, 1998; Topcupglu et al., 2005; Cunha and Silva, 2007; Chen, 2007; Silva and Cunha, 2009; Filipović et al., 2009; Wang et al., 2010).

However, most studies simplify the interhub cost function as a linear function which is independent of flows. Zangwill (1968) noted that the linear cost assumption is often not realistic while considering in the certain network problems. Zangwill found out a theorem established that for either single source or single destination networks the multi-commodity can be transformed to the single
commodity case. Zangwill also used the characterization of extreme points to determine the minimum concave cost solution for such single commodity networks. Klincewicz (1990) proposed an algorithm for solving a freight transport problem (FTP). In this case, the freight transport problem is decomposed into a set of concave cost facility location problems (CFLP). Moreover, the piecewise linear concave function is used to account for the different discounts corresponding to different flow volumes.

As O’Kelly et al. (1996) observed the different discount factors settled would result the different hub selections and non-hub allocations. They also mentioned that the optimal solution of multiple allocations could be a lower bound for the single allocation problem. O’Kelly and Bryan (1998) pointed out that the current model of oversimplifying the interhub travel cost not only miscalculates the total transportation cost, but may also erroneously select optimal hub locations and allocations. Hence, they developed a formulation model, called FLOWLOC, in which the interhub travel costs are non-linear cost functions depending on the flows between hubs. A set of piecewise linear functions is chosen to approximate the non-linear cost function.

Tackling the FLOWLOC, Bryan (1998) extended the FLOWLOC to four different models: (1) a capacitated network model; (2) a minimum threshold model; (3) a model that endogenously determines the number of hubs; and (4) a model that incorporates a flow-dependent cost function for the spokes as well as the interhub links. Klincewicz (2002) presented a model which is transformed the FLOWLOC to the uncapacitated facility location problem (UFLP) if the set of hubs is fixed. To figure out the effects with the different sets of piecewise linear function and the hub locations, Klincewicz used the well-known CAB data set to be a testing bed and proposed an enumeration procedure and two heuristics based on tabu search (TS) and greedy random adaptive search procedures (GRASP). Kimms (2006) also extended the FLOWLOC model to three conditions in which the economies of scale are considered on all kinds of connections: (1) quantity discounts; (2) fixed costs; and (3) multiple modes. Kimms also illustrated a small example for each condition and solved them by CPLEX. Camargo et al. (2009) presented a new and tighter formulation based on the FLOWLOC for the uncapacitated multiple allocation hub location problem (UMAHLP). Two versions of benders decomposition, cost- and service-driven, are proposed. The results obtained by these two versions of benders decomposition are better than the FLOWLOC both on the efficiency and solution quality of upper- and lower-bounds.

Racunica and Wynter (2005) formulated a nonlinear model with piecewise concave functions on interhub connections as well as on hub-to-destination connections. They proposed two heuristics for solving a piecewise approximation of the intermodal freight transport problem. Cunha and Silva (2007) proposed a genetic algorithm (GA) for solving the USAHLP, and then applied in a less-than-truckload service with a flow-dependent concave cost function.

There are a great number of algorithms to tackle the concave cost transportation network problems. However, as the pioneering work of O’Kelly and Bryan (1998), the flow-dependent cost function is a new issue in the H&S network literature. A set of piecewise concave functions has been approximated to the nonlinear function. Unfortunately, lack of studies addresses the relative formulation for the USAHLP. Thus we develop a model and propose a threshold accepting (TA) algorithm to fill a gap in this topic.

This remainder of the paper is organized as follow. In Section 2 we provide a general description and a formulation model for our problem. Section 3 presents our algorithm structure and several computational results are provided in Section 4. Finally, the remarks and directions for further research are discussed in Section 5.

2. MATHEMATICAL FORMULATION

2.1 Problem Description

Underlying description is given before we define the mathematical model: Let \( N \) be the set of nodes that exchange traffic and \( d_{ij} \) be the Euclidean distance between node \( i \) and node \( j \). There are \( W_{ij} \) (\( \geq 0 \)) units of flow that must be routed from origin \( i \) to destination \( j \) via one hub \( k \) or two hubs \( k \) and \( l \) \((i,j,k, l \in N)\). That is, if one hub is used, the traffic \( W_{ij} \) flows on the link path \( i-k-j \). Otherwise, the flow is routed via an interhub connection on the link path \( i-k-l-j \) whereas \( k \neq l \). Furthermore, an installation cost of establishing a hub at node \( k \) is represented as \( F_k \).

The costs per unit of flows, the collection cost \( \lambda \times d_{ik} \) and the distribution cost \( \delta \times d_{lj} \), are assumed on the links \( i-k \) and \( l-j \), respectively. The cost on the interhub link \( k-l \) is a non-linear concave function of the total flows on that link, as proposed by O’Kelly and Bryan (1998). Like Klincewize (2002) noted, a piecewise linear cost function (showed as Figure 1) can be approximated and viewed as the “lower envelope” of a set \( Q \) of linear functions, where each segment \( q \in Q = \{1, 2, 3, \ldots \} \) is one of piecewise lines. Each function can be described in terms of an intercept \( b_{q_{ikl}} \) and a slope \( \alpha_{q_{ikl}} \), where each intercept follows the equation (1). Therefore, the total cost on the interhub connection \( k-l \) is given by the linear function that minimizes \( (b_{q_{ikl}} + \alpha_{q_{ikl}}X_{q_{ikl}}) \times d_{kl} \), where \( X_{q_{ikl}} = 1 \) if the bundled flow uses the path \( i-k-l-j \), and 0 otherwise.

\[
b_{q_{ikl}} = (\alpha_{q_{ikl}} - \alpha_{q_{ikl}}) \times S_{q_{ikl}} + b_{q_{ikl}}
\]  

\( (q_{ikl}) \) (1)
where \( S_{qij} \) is the flow threshold corresponding to the linear function \( q \). \( Z_{qij} = 1 \) if the arc flow between hubs \( k \) and \( l \) uses segment \( q \) and 0 otherwise; \( R_{qij} \) is the total flow when segment \( q \) is used; \( Y_{ik} = 1 \) if node \( i \) is assigned to \( k \) and 0 otherwise.

\[
\sum_{k \in N} X_{qij} = Y_{ik} \quad \forall i, j, k \in N \tag{10}
\]

\[
X_{qij} \in \{0, 1\} \quad \forall i, j, k, l \in N \tag{11}
\]

\[
R_{qij} \geq 0 \quad \forall q \in Q; k, l \in N; k \neq l \tag{12}
\]

\[
Z_{qij} \in \{0, 1\} \quad \forall q \in Q; k, l \in N; k \neq l \tag{13}
\]

\[
Y_{ik} \in \{0, 1\} \quad \forall i, k \in N \tag{14}
\]

The objective function (2) minimizes the total transportation costs. Constraint (3) computes the amount of traffic on the interhub link \( k-l \). Constraint (4) ensures that this interhub traffic is correctly associated with the corresponding slope \( a_{qij} \) and the intercept \( b_{qij} \). Constraint (5) guarantees that each interhub link \( k-l \) is utilized, i.e., ensuring at least the interacting traffic between hubs \( k \) and \( l \) is routed on this link. Constraint (6) forces that exactly one segment \( q \) of piecewise linear function is used on each interhub link. Constrain (7) states that every node is assigned to exactly one hub. Constrain (8) ensures that node \( i \) can be allocated to hub \( k \) only when \( k \) is selected as a hub. Constraint (9) requires that for every destination \( j \), the total flow from origin \( i \) to destination \( j \) routed via paths using link \( i-k \) will be nonzero only if node \( i \) is assigned to hub \( k \). Similarly, constraint (10) assures that for every origin \( i \) and every hub \( k \), a flow through the path \( i-k-l-j \) is feasible only if \( j \) is allocated to hub \( l \). Constraints (11)-(14) are binary integrality and non-negative constraints, respectively.

3. PROPOSED THRESHOLD ACCEPTING ALGORITHM

The threshold accepting (TA) algorithm which is a variant of the classical simulated annealing (SA) algorithm, was first introduced by Dueck and Scheuer (1990). In TA, a simpler acceptance criterion, so-called threshold, is adopted for accepting a worse solution rather than the probabilistic acceptance in the SA. Threshold is a deterministic value to decide whether or not to accept the worse solution. If the new configuration is not much worse than the current one within the threshold, then TA accepts the new configuration as current solution. The threshold is reduced based on a specific rule during the solution search and gradually decrease to 0 in the end. The keystones of TA are the specific types of lowering threshold, stopping criteria as well as the schemes for generating initial and neighborhood solutions. As shown in its simplicity, TA has been implemented on different combinatorial problems with excellent performance.
3.1 Hierarchical TA

Considering the attributes of USAHLP, our TA consecutively solves a series of uncapacitated single allocation p-hub median problem (USApHMP) by adopting the add-policy. The idea is that: Let X be the set of all feasible solutions of the problem. TA proceeds in a hierarchical manner, searching for the best solution. To find the global best solution so far, Xₖ by searching transitions. For every moving operation in each iteration, the algorithm decides the acceptance between the new solution X’ and the current one Xₖ based on the threshold and the replacement of the corresponding best solution Xₖ. To find the global best solution Xₙ from each different number of hubs, we intend to repeatedly execute TA by increasing the number of hubs until the corresponding objective function value of the best solution of p, f(Xₖ), is larger than that of the solution for p – 1. However, as Zangwill (1968) addressed, the characterization of extreme points is an explicitly useful feature for certain types of concave cost networks. A modification for the hierarchical TA (Wang et al., 2010) is proposed for USAHLP with concave cost function. Throughout our TA procedures, we design an estimation mechanism to roughly approximate the slope b_{ijkl} for every interhub link k-l. The concept of estimation mechanism is in section 3.2. The algorithms for our TA are as follows:

- **Step 1:** Let p (number of hubs) start from 1. Generate the initial solution X₀, let X₁ = X₀, and f(X₁) = f(X₀), Setup the decreased factor (γ), 0 < γ < 1.

- **Step 2:** Set up the initial threshold value (TH₀), the length of threshold string (L), and the moving length (M). Let Xᵢ = X₀, f(Xᵢ) = f(X₀), i = 1, TH(i) = TH₀.

- **Step 3:** Set j = 1.

- **Step 4:** Generate the neighborhood solution (X’) from Xᵢ.

- **Step 5:** If p = 1, using the following equation to obtain the objective function value, f(X’) = Σ_{i≠k}Σ_{j≠k}W_{ij}(λ d_{ik} + δ d_{ij})X_{ijkl} + Σ_{i≠k}F_{ij}Y_{ik}; otherwise, using the slope b_{ijkl} obtained by the estimation mechanism to calculate the objective value f(X’).

- **Step 6:** Calculate ΔE = f(X’) - f(Xᵢ), if ΔE ≤ TH(i), go to step 7; otherwise, go to step 9.

- **Step 7:** Accept the neighborhood solution, let Xᵢ = X’ and f(Xᵢ) = f(X’).

- **Step 8:** If f(X’) < f(Xᵢ), let Xᵢ = X’ and f(Xᵢ) = f(X’).

- **Step 9:** If j < M, j = j + 1 and go to step 4; otherwise, go to step 10.

- **Step 10:** If i < L, then TH(i+1) = γ × TH(i), i = i + 1, go to step 3; otherwise, go to step 11.

- **Step 11:** If f(Xᵢ) < f(Xᵢ), let Xᵢ = Xᵢ and f(Xᵢ) = f(Xᵢ), and increasing the number of hubs by 1 (p = p + 1). Generate a new initial solution X₀ based on Xₚ, go to step 2; otherwise, go to step 12.

- **Step 12:** Do the local search and output the global best solution Xₙ and f(Xₙ).

A simple rule for generating the initial solution is adopted in our TA implementation. A node with the higher rank, which is measured by the total incoming and outgoing flows for the node divided by its installation cost, will be chosen as a hub at the beginning. The rationale of this idea is to locate a hub that can handle the maximum flow per unit cost in order to serve all flows with less installation costs. Then, a nearest-distance base assignment is applied for allocating all non-hub nodes. The following types of transitions are used to generate neighborhood solutions.

- **Type 1:** When number of hubs p is equal to 1, randomly choose a non-hub node to replace the hub node and reassign all non-hub nodes to the new hub.

- **Type 2:** If a cluster consists of only the hub, then perform the following move.
  - Pick randomly a non-hub node and make it to a hub.
  - Reallocate non-hub nodes to its nearest hub.

- **Type 3:** If the moving length of transitions reaches the pre-defined fixed length then do this exchange to generate the new solution; otherwise, goes to type 4 or 5 depending on a pre-set probability.
  - Randomly pick a hub and non-hub node, and exchange each other.
  - Reallocate non-hub nodes to its nearest hub.

- **Type 4:** Change the allocation of a randomly chosen non-hub node to one of the nearest p or three (if p > 3) hubs.

- **Type 5:** Change the location of the hub within a randomly chosen cluster to a different randomly chosen node in the cluster.

3.2 Estimation Mechanism

There are a bundle of flows from nodes i and j that must be transferred via interhub links when the number of hubs p is larger than or equal to 2. Hence, the estimation mechanism is launched to result a rough approximation for obtaining the discount factor α (i.e., the slope b_{ijkl}). First, an average flow for hubs, which is approximated to the total flow on interhub connection k-l (R_{ijkl}), is obtained from dividing the total network flow by the current number of hubs p. Then, we can find the corresponding slop α_{ijkl} and intercept b_{ijkl} from the piecewise linear concave function. Our TA is going to calculate the objective function value...
(eq. (2)) using this estimation result throughout the current searching process until the number of hubs $p$ increases. An example is illustrated as follow: Assuming the current number of hubs $p$ is 2, and the total network flow is equal to 350. Subsequently, the estimated average flow is resulted by 175 per hub handled. Finally, as our estimation scheme, the slope $a_{gi}$ and intercept $b_{gi}$ are 0.6 and 25 corresponding to the piecewise linear function in figure 2, respectively. This piecewise arc cost function will be used for computing the interhub link.

![Figure 2: An example for the estimation scheme](image)

4. COMPUTATIONAL EXPERIMENTS

In this section, the empirical results of TA are given. To evaluate our proposed TA, two data sets, CAB and AP, are tested as benchmarks and each instance is run for 20 times. Both data sets can be downloaded from OR-Library. In order to examine the validities of our model and results, a commercial software CPLEX was applied. Both CPLEX and our TA were coded in C++ by Microsoft Visual Studio .NET 2003 and carried out on an Intel(R) Core 2 Duo 2.66 GHz CPU, with 2 GB of RAM, running under Windows XP Professional. Furthermore, two different piecewise linear concave functions with four slopes used in Klincewicz (2002) and Camargo et al. (2009) were adopted in Table 1. Each instance was solved for 20 times. To result a stable performance for TA, a series of preliminary experiments is conducted. A set of control parameters is suggested as Table 2.

The definitions of headings used in tables are as follows in the tables of our computational results:

- Min – the gap (%) between the best found solution by TA and the optimal solution;
- CPU – total running time in seconds;
- Opt - the optimal solution obtained by CPLEX

- BKS – best known solution by our experiment;

<table>
<thead>
<tr>
<th>Table 1: Piecewise linear concave functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>flow on $k/l$ link (x1,000)</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
<tr>
<td>$0 \leq R_{pl} &lt; 50$</td>
</tr>
<tr>
<td>$50 \leq R_{pl} &lt; 100$</td>
</tr>
<tr>
<td>$100 \leq R_{pl} &lt; 200$</td>
</tr>
<tr>
<td>$200 \leq R_{pl}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: A set of control parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>$n \leq 50$</td>
</tr>
<tr>
<td>$n \geq 100$</td>
</tr>
</tbody>
</table>

4.1 Results for CAB Data Set

The CAB (Civil Aeronautics Board) data set was first introduced by O’Kelly (1987). It is a subset, in which includes 25 cities, of a larger data consisting of the flows between 100 U.S. cities in 1970. The distance and flow between any pair of cities are symmetric, where the former satisfies the triangular inequalities and the later accounts for 51% of the flows observed from 100 cities. The data set contains 48 instances with up to 25 nodes, and the installation costs of nodes for being a hub are equally set to be 100, 150, 200 and 250. Further, the unit costs on the collection $\lambda$ and distribution $\delta$ are equal to 1.

The results obtained from CPLEX and our TA are presented in Table 3. The first two column headings are the number of nodes and the installation cost, respectively. Columns 3-6 are the optimal solution and computational times for cost function 1 and 2, respectively. The gap for the best found solution and CPU time for functions 1 and 2 by our TA are in columns 7-10 as well. As the results showed, the network enforces the traffic routed through interhub links with lower discount while the installation cost is fixed, especially when the number of nodes increases. Moreover, the computational time for CPLEX is dramatically increased with the number of nodes increases, and is decreased when the installation cost raise. Our proposed TA reaches the optimal solution for each instance. Regarding the computational time, our TA can find the optimal solution with much shorter time and the number of nodes has less effect in the efficiency.

4.2 Results for AP Data Set

The AP (Australia Post) data set introduced by Ernst and Krishnamoorthy (1996) encloses 200 nodes along with
their coordinates and volumes of mail flows, in which the postcode districts in Australia are presented. The largest initial instance can separate an amount of small instances with 10, 20, 25, 40 and 50 nodes by aggregating the mail flows. The cost coefficients for the collection \( \lambda \) and contribution \( \delta \) are 3.0 and 0.75. Regarding the installation cost, AP includes two types, the tight and the loose, of unequal installation cost for each node. A difference between CAB and AP is that the mail flows are transferred within each city and non-symmetric. Moreover, Silva and Cunha (2009) introduced four new larger instances with 300- and 400-node based on AP for pressing close to the practice. The way they used for generating is described in Silva and Cunha (2009). The results are given in following tables.

The results obtained by CPLEX and our TA for the loose type are presented in Table 4. The first two column heading are the piecewise linear concave function used and number of nodes, respectively. Columns 3-6 are the best known solution, CPU time by CPLEX and minimum gap and CPU time by TA, respectively. If the CPLEX cannot solve the instance, a "-" is shown in the CPU. According to the limitation of the number of constraints, only small instances with 10, 15 and 25 nodes can be solved by CPLEX. The best known solution for \( n > 25 \) is the best among 20 runs by TA. Our TA can obtain the optimal solution for small size instances in both cost functions. The CPU time is much smaller than that of CPLEX. The CPU time by our TA increases as the instance size increases.

Table 5 shows the results for the tight type of AP data. The column headings are the same as those in table 4. As mentioned by Ernst and Krishnamorthy (1999), tight instances, which those nodes with high traffic are set with higher installation costs, are harder to solve. Therefore, we can observe that the CPU times for the tight type instances are higher than those for the loose types at the same number of nodes. From Table 5, our TA performs the same
behavior as that of loose type. Our TA reaches the optimal solution for small instances. Furthermore, our TA searches the optimal solution/best solution within 21.48 seconds for the average CPU.

<table>
<thead>
<tr>
<th>Fun.</th>
<th>n</th>
<th>BKS</th>
<th>CPU</th>
<th>Min</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>271208.26</td>
<td>4.30</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>20</td>
<td>1146.08</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>18974.60</td>
<td>0.00</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>293164.84</td>
<td>-</td>
<td>0.00</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>300420.99</td>
<td>-</td>
<td>0.00</td>
<td>0.16</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>305979.95</td>
<td>-</td>
<td>0.00</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>276535.52</td>
<td>-</td>
<td>0.00</td>
<td>80.46</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>235483.63</td>
<td>-</td>
<td>0.00</td>
<td>21.11</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>229557.28</td>
<td>-</td>
<td>0.00</td>
<td>91.29</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>271208.26</td>
<td>3.28</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>20</td>
<td>1438.33</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>6783.67</td>
<td>0.00</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>293164.84</td>
<td>-</td>
<td>0.00</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>300420.99</td>
<td>-</td>
<td>0.00</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>305979.95</td>
<td>-</td>
<td>0.00</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>276535.52</td>
<td>-</td>
<td>0.00</td>
<td>80.22</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>235483.63</td>
<td>-</td>
<td>0.00</td>
<td>21.15</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>229557.28</td>
<td>-</td>
<td>0.00</td>
<td>91.49</td>
<td></td>
</tr>
</tbody>
</table>

*: the optimum obtained by CPLEX

Based on the results for both CAB and AP data set, our proposed TA, which contains an estimation method, is a good algorithm for the USAHL problem with piecewise linear concave cost function on the interhub arcs. O’Kelly et al. (1996) mentioned that the solution of multiple allocation problem could be considered as a lower-bound for single allocation problem. To further analyze the validity of our results, an evaluation could be adopted to quantify the gap between our result and the lower-bound.

5. CONCLUSIONS

In H&S networks, cost savings occur on links, especially on interhub links, that are able to bundle and transship traffic. This paper works on a well-known problem—USAHL problem—with concave function. A model is introduced for accounting for scale economies by allowing interhub costs to be a flow-dependent function—as flows increase, costs increase at a decreasing rate. However, as literature reviewed, a piecewise linear approximation is fitted well and shown to be a good substitute for the nonlinear cost function. To solve this problem in a reasonable time, we extend our previous TA skeleton with an estimation mechanism for the piecewise linear concave function.

From the empirical tests, we observe that our TA algorithm has a steady performance for each instance on solution quality and less effects of the number of nodes on computational efficiency. Our TA can provide a good result for the largest instances. In the future, we could compare our single allocation results with those of multiple allocation to obtain the gap with the lower bound. Another extension is to study the problem with capacitated constraint on the hubs which is closer to the real-life situation.

REFERENCES


**AUTHOR BIOGRAPHIES**

Hung-Jie Wang is a manufacturing engineer at Powerchip Technology Corporation, Hsinchu, Taiwan. He received his M.S. and B.S. degree in Industrial Engineering and Management Department from the Yuan Ze University. His primary research interests are in logistics management and metaheuristic algorithms. His email address is hjwang@powerchip.com.

Ching-Jung Ting is an associate professor in the Industrial Engineering and Management Department at Yuan Ze University, Taiwan. He received his B.B.A. from National Chiao Tung University, Taiwan, M.S. in Civil Engineering from Northwestern University, and Ph.D. in Civil Engineering from University of Maryland, College Park, respectively. His primary research interests include supply chain management, logistics management, metaheuristics, and transportation system analysis. His email address is ietingcj@saturn.yzu.edu.tw.