# A chartered bus allocation problem

Woo-Je Kim

Department of Industrial Engineering, Daejin University,

San11-1 Sundanri, Pochun, Kyungkido 487-711, Korea

phone:+82-31-539-2001; fax:+82-31-539-2000; wjkim@road.daejin.ac.kr

Katta G. Murty

Department of IOE, University of Michigan,

Ann Arbor, Michigan 48109-2117, USA

phone:734-763-3513; fax:734-764-3451; katta_murty@umich.edu

October 28, 2000

**Abstract**

We present a case study dealing with optimizing the allocation of buses to trips requested by customers at a chartered bus rental company. It is a multiobjective problem. We model the problem using a graph, and develop an algorithm for solving it, including a heuristic approach for meeting an objective which is stated in the form of a goal. Lower bounds for objective values indicate that our approach yields excellent results. In addition, tests on company data over several days show that our approach leads to solutions which reduce one of the important objective functions by over 10% from solutions obtained manually at the company.

*Keywords*: Chartered bus allocation problem; Dilworth's minimal chain decomposition;

## 1 Introduction

A chartered bus company (Arirang in Seoul, South Korea) rents buses along with drivers (the driver of each bus is solely responsible for driving it) to customer service requests. Their customers are typically groups of people who need a bus for a trip starting at a specified location at a starting time, and taking them to a desired ending location by an ending time (the trip might involve brief stops along the way, but the driver has to be at the service of the group continuously between the starting time and ending time).

Some groups (typically groups of sightseeing tourists) want to be taken from an origin location to a destination location in the morning, and after several hours, to be picked up at that destination location, and to be brought back to the starting location. Since the group does not need the bus in the interval between the forward and return parts of this roundtrip, and since the forward and return parts of this roundtrip can very well be carried out by two different buses, we will consider those forward and return parts as two separate trips.

The problem considered in this case study is concerned with allocating buses to such service requests which we will call **trips** or **tasks** in the sequel. The company has two types of buses, a 45 seat bus (20 available) used for large groups and a 15 seat bus (5 available) used for small groups, stationed at two depots in separate locations in the city. The company can itself borrow additional buses of each size on a daily basis from outside vendors if there is a need for them, and staff them from its pool of spare drivers.

Each trip has the following data associated with it: size of vehicle needed (depending on customer group size), starting time and location of the trip, ending time and location of the trip. The **duration of a trip** which is the difference between its ending and starting times varies between 0.5 hours to 20 hours, but more than 75% of the trips have duration $\leq$ 5 hours. We use the following notation:

$n$ = total number of trips to be handled on a day

$\underline{t}_i$, $\overline{t}_i$ = starting and ending clock times of trip $i$, for $i = 1$ to $n$

$p_i$, $q_i$ = starting and ending locations of trip $i$, for $i = 1$ to $n$

The company usually handles about 50 to 80 such trips per day. The data on 68 trips requiring the larger type of buses on one day is given in the appendix as an illustrative example. In this data all the locations are numbered serially, and for each trip we mention the numbers of the starting and ending locations. This data may be useful to faculty members to construct a computational project for their classes.

By making small perturbations in the starting times of the trips if necessary, we assume that their values for the various trips are distinct. Then we number the trips serially in increasing order of their starting times.

After completing a trip with ending time $\overline{t}_i$, a bus can take up another trip whose starting time is after $\overline{t}_i$. In the same way a bus may handle a subset of trips one after the other during the day.

Suppose a bus handles trips numbered $i_1, \cdots, i_r$ in that order on a day. This sequence of trips $(i_1, \cdots, i_r)$ is called that bus's **worksequence** for that day. Then for each $s = 1$ to $r - 1$ after completing trip $i_s$ at clock time $\overline{t}_s$ at location $q_s$, the driver of that bus has to drive to the starting location $p_{s+1}$ of the next trip $i_{s+1}$ before its starting time $\underline{t}_{s+1}$. So the following condition

$$\underline{t}_{s+1} - \overline{t}_s \geq \text{driving time from } q_s \text{ to } p_{s+1} \tag{1}$$

must hold for all $s = 1$ to $r - 1$ for $(i_1, \cdots, i_r)$ to be a worksequence. In this case, the drives from $q_s$ to $p_{s+1}$ for $s = 1$ to $r - 1$ of this bus are called **empty load drives** on this worksequence. During an empty load drive the company is incurring the cost of keeping the bus running (fuel etc. + driver's wages) on its own without any customer paying for it.

If the inequality in (1) holds as a strict inequality, then after reaching location $p_{s+1}$ and parking the bus, the driver has to wait there idly until clock time $\underline{t}_{s+1}$ when trip $i_{s+1}$ starts. This idle time is called **waiting time**. During this waiting time the company incurs the cost of the driver's wages on its own without receiving any income from customers to offset it.

All the buses start at their depot and return to their depot after their last trip in their worksequence for the day. So for this bus the quantity

$$\overline{t}_r - \underline{t}_1 + \text{(driving time from depot to } p_1) + \text{(driving time from } q_r \text{ to depot)}$$

measured in hours represents the time in hours the driver of that bus worked that day, and this quantity is called the **duration of the worksequence** $(i_1, \cdots, i_r)$ (note that this depends on the depot of the bus to which this worksequence is assigned). The driver's wages for that day will be proportional to the duration of the worksequence assigned to him/her. For this reason most drivers have a strong desire to maximize the duration of their worksequences each day, even though for the sake of safety we need to make sure that the duration of worksequences used are not unreasonably large.

The **maximum duration constraint** requires that duration of worksequences used should be $\leq$ a **safety limit** (currently 13 hours). However, this is not a hard constraint that is enforced on every worksequence because in many long duration worksequences there are waiting time intervals and other nondriving time intervals within trip durations during which the driver can rest and snooze and refresh his body. The company's policy is that the percentage of worksequences of duration greater than or equal to the safety limit should be kept below 50% as far as possible. The reason for setting this 50% as a desirable upper limit for the percentage of long worksequences is the following: the company tries to assign a short worksequence on a day to any driver who handled a long worksequence the previous day. Keeping the percentage of long worksequences below 50 makes it possible to balance a long worksequence with a short one the next day.

The problem is to partition the set of trips into worksequences for the buses. The company has to solve this problem every evening for the set of trips they need to handle the next day, which consists of about 50 to 80 trips.

## 2 Separating long duration trips

Trips with duration $\geq$ 11 hours are called **long duration trips**, typically, less than 10% of the trips are of this type. Since the driver of a bus assigned to a trip is required to be with the customer during the entire duration of the trip, any driver assigned to such a long duration trip is committed to at least 11 hours of work on that day, and cannot be assigned any other work. So each of these long duration trips is accepted as a single trip work sequence in the solution, and taken out of further consideration.

The company has the policy of making sure that these and any other long duration worksequences adopted are distributed equally among all its drivers as far as possible. This assignment of worksequences to drivers is carried out manually. They also try to make sure that the percentage of days on which drivers are assigned long duration worksequences is kept below 33% as far as possible.

## 3 The Network $G$ for modeling the problem

We represent each trip $i$ by a node $i$ in a directed network. We include an arc $(i, j)$ from node $i$ to node $j$ if a bus can handle trip $j$ after handling trip $i$ (the condition for including this arc is : (driving time from $q_i$ to $p_j$) + $\bar{t}_i \leq \underline{t}_j$). We will, however, have nodes corresponding to long duration trips (those with duration $\geq$ 11 hours as defined in Section 2) as isolated nodes without any arc incident into or out of them. The reason for this is, as mentioned in Section 2, these trips are already too long to constitute a full day's work for a driver, that we will not consider combining them with any other trip in the worksequence assigned to a driver. Let the resulting network be $G = (\mathcal{N},\mathcal{A})$ where $\mathcal{N}$ = set of nodes = $\{1, 2, \cdots, n\}$ and $\mathcal{A}$ = set of arcs constructed as above.

A similar network was used in [2] to model vehicle scheduling problems in which vehicles must be assigned to time-tabled trips. We will use the same network to model our problem.

By the manner in which trips are numbered, it is clear that if $j > i$, then $(j, i) \notin \mathcal{A}$. This implies that the numbering of the nodes in $G$ is an acyclic numbering (i.e. all arcs go from a node to another node with higher number), and hence $G$ is an acyclic network.

We define a simple chain in $G$ to be either a set containing a single node, or a sequence of more than one node $i_1, i_2, \cdots, i_k$ satisfying the condition that $(i_{r-1}, i_r) \in \mathcal{A}$ for $r = 2$ to $k$. Thus it corresponds to the usual notion of a simple chain in network terminology [11] when there are two or more nodes in it. However, a single node by itself is also considered as a simple chain (it has no arcs) in this context. Therefore, every worksequence corresponds to a simple chain in $G$ and vice versa.

In the network $G$, each arc $(i, j)$ represents the opportunity of a bus servicing trip $j$ after servicing trip $i$. If this happens, then that bus, after finishing the servicing of trip $i$ at clock time $\bar{t}_i$, travels from $q_i$ to $p_j$, suppose this travel time is $d_{ij}$ hours, and then the bus has to wait $\underline{t}_j - \bar{t}_i - d_{ij} = w_{ij}$ hours at $p_j$ before starting the servicing of trip $j$ at clock time $\underline{t}_j$. Thus $d_{ij}$ hours is the travel time for empty load drive, and $w_{ij}$ hours is the waiting time associated with arc $(i, j) \in \mathcal{A}$. We use the following dollar conversion values for these times

| | | |
|---|---|---|
| one hour of driver's waiting time | = | \$30 (includes driver's salary from the company + lost opportunity for bus to make profit), |
| one hour of time spent in empty load drive | = | \$40 (includes driver's salary from the company + lost opportunity for bus to make profit + cost of fuel, maintenance, etc to keep bus running during the hour). |

Using these conversion values, the cost coefficient of arc $(i, j) \in \mathcal{A}$ becomes $c_{ij} = 30w_{ij} + 40d_{ij}$.

# 4 The multiobjective nature of the problem

The company puts the highest priority in handling all the trips using the smallest number of buses each day. This leads to the problem of minimizing OBJ1 = the number of worksequences into which the set of trips is partitioned.

Next in order of importance is minimizing OBJ2 = cost of empty load drives and waiting times in all the worksequences adopted.

A third objective is to keep OBJ3 = the percentage of worksequences violating the maximum hours of work constraint below 50 as far as possible.

## 5 Approaches for solving the problem

The problem considered belongs to the well-studied class of full-truckload problems in vehicle routing and scheduling [1, 2, 3, 4, 8, 9, 10, 12]. When there is only a single objective function to optimize, it can be solved using a column generation technique very similar to that used to solve airline crew-scheduling problems [13]. However, the multiobjective nature of our problem, and in particular OBJ3 which is in the form of a goal rather than an objective function to be optimized, the two types of buses for large and small groups, the multiple depots, and the possibility of renting additional buses from outside vendors, make it very difficult to adopt column generation techniques to solve our problem. For this reason we did not pursue column generation techniques to solve our problem, instead we used the direct approach based on minimal chain decomposition of the network $G$ discussed in Section 3. The application of this approach to other vehicle routing problems has been discussed in [5].

To make the application of Dilworth's minimal chain decomposition to our problem easier, we will split OBJ2, the cost of empty load drives and waiting times into two parts as follows: OBJ2 = OBJ2.1 + OBJ2.2, where OBJ2.1 = cost of empty load drives and waiting times in-between consecutive trips on worksequences, and OBJ2.2 = cost of empty load drives from the depot to the starting location of first trip and from ending location of last trip to the depot; in all the worksequences adopted. We will handle the task of minimizing OBJ2 in two stages. First we will find a partition of the trips into a minimum number of worksequences having minimum value for OBJ2.1. Once these worksequences are determined, we will find an allocation of buses to each of these worksequences that minimizes OBJ2.2.

## 6 How to handle the two types of buses?

As mentioned in Section 1, the company has two types of buses, 15 seat and 45 seat ones. Ideally they would like to use the 15 seat buses to service trips for small groups (with 15 or less people), and 45 seat buses for trips with larger groups. However on days when there is a large number of trips with smaller groups, the company may experience a shortage of 15 seat buses to handle all of them. On such days, instead of renting some extra 15 seat buses from outside vendors, the company has found it to be much more economical to assign some of its own 45 seat buses to trips with small groups. Because of this policy, we use the following procedure for handling the allocation of two size buses to the various trips. The approach for partitioning the set of trips into worksequences discussed in the next section assumes that all the trips can use one size buses. The approach is applied to our problem in the following way:

i) First consider only the small group trips for which the 15 seat bus is suitable. Apply the approach discussed in the next section to partition this subset of trips into worksequences. Find the total working time associated with each of the worksequences obtained. Some of these worksequences may correspond to very small durations, others may correspond to working times that form a reasonable day's work for a driver. Select a threshold value, say $\delta$ hours, as a lower bound for a day's working time. For all worksequences associated with a working time $\geq \delta$, assign 15 seat buses to the extent they are available. We used $\delta = 6$ hours.

ii) The trips on all the worksequences associated with working time $< \delta$, and all the other worksequences for which 15 seat buses may not have been allocated in i) are combined with the set of large group trips. We then apply the approach discussed in the next section to partition this set of trips into worksequences for each of which we consider allocating one of the larger size buses.

On rare occasions, they get service requests involving groups of more than 45 people. Such a request cannot obviously be handled by any one bus. The company breaks such large groups into smaller subgroups each of which can be handled by one of the available buses. In this case, the request of each of these subgroups is treated as a separate trip before applying the approach.

# 7 The Model to partition the set of trips into worksequences

Here we discuss a model for partitioning the set of trips into the smallest number of worksequences that simultaneously minimizes OBJ2.1 (which is the cost of empty load drives and waiting times in-between consecutive trips on worksequences) among all such minimal partitions.

Since each worksequence corresponds to a simple chain in the acyclic network $G$ defined in Section 3 and vice versa, the problem of partitioning the set of trips into the smallest number of worksequences in the same as that of finding a minimum cardinality simple chain cover for all the nodes in $G$, which is known in Network Programming literature as **Dilworth's minimal chain decomposition problem**. An efficient algorithm for it based on the maximum cardinality bipartite matching algorithm has been developed by Fulkerson[6] and discussed in Ford and Fulkerson[5] (See also Murty[11]). The algorithm involves finding a maximum cardinality matching in the bipartite network $B = \{\mathcal{N}_1, \mathcal{N}_2; \mathcal{A}_1\}$ with node set $\mathcal{N}_1 = \{R_1,\cdots,R_n\}$, $\mathcal{N}_2 = \{C_1,\cdots,C_n\}$ and edge set $\mathcal{A}_1 = \{(R_i, C_j) : (i, j)$ is an arc in $\mathcal{A}$ in $G\}$ where $n$ is the number of nodes in $G$ (equal to the number of trips). Suppose the cardinality of a maximum cardinality matching in $B$ is $r$. Then it is shown that the minimum number of simple chains needed to cover all the nodes in $G$ is $n-r$; i.e. in our problem at least $n-r$ worksequences or buses are needed to cover all the trips. From any maximum cardinality matching $M$ in $B$, an easy procedure is available for deriving a set of $n-r = n-|M|$ simple chains in $G$ to cover all the nodes (See [5, 11]). This procedure consists of obtaining the set of edges $\bar{\mathcal{A}} = \{(i, j) : (R_i, C_j)$ is an edge in the matching $M\}$ in $G$. Then $(\mathcal{N}, \bar{\mathcal{A}})$ decomposes into a node disjoint collection of simple chains in $G$, this collection of simple chains is a minimum cardinality simple chain cover for the nodes of $G$. The sequence of trips corresponding to nodes in the order in which they appear on each of these simple chains is a worksquence for a bus, and hence each of these simple chains can also be interpreted as a bus route.

The maximum cardinality matching problem in $B$ usually has many alternate optimum solutions and any one of them can be used to get a minimum cardinality simple chain cover for the nodes in $G$ having $n-r$ simple chains. Define, for $i, j = 1$ to $n$

$$x_{ij} = \begin{cases} 1 & \text{if } (R_i,\, C_j) \text{ is in the selected matching,} \\ 0 & \text{otherwise.} \end{cases}$$

If $x_{ij} = 1$, i.e., $(R_i,\, C_j)$ is a matching edge in the matching selected, then arc $(i,\, j)$ in $G$ will be an arc in one of the simple chains in the simple chain cover corresponding to the selected matching and vise versa. Hence the objective function $Z(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij}$ gives the value of OBJ2.1 in the partition of the set of trips into worksequences corresponding to the matching $x = (x_{ij})$. Therefore, the problem of finding a minimum cost (OBJ2.1) minimum cardinality partition of the set of trips into worksequences is the following:

$$\begin{aligned} &\textit{Minimize} \quad && Z(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ &\textit{subject to} && \sum_{i=1}^{n} x_{ij} \leq 1 \\ & && \sum_{j=1}^{n} x_{ij} \leq 1 \\ & && \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} = r \\ & && x_{ij} = 0 \text{ if } (R_i, C_j) \text{ is not an arc in } B. \\ & && x_{ij} \in \{0, 1\} \text{ if } (R_i, C_j) \text{ is an arc in } B. \end{aligned}$$

Since $r$ is the cardinality of a maximum cardinality matching in $B$, this is the problem finding a minimum cost maximum cardinality matching in the bipartite network $B$. Every extreme point optimum of the LP relaxation of this problem will be integer, and hence it will be the incidence matrix of a minimum cost maximum cardinality matching in $B$. So find an optimum extreme point solution of the LP relaxation of this problem, $\overline{x} = (\overline{x}_{ij})$. Let the set of edges in it $\{(i,\, j) : \overline{x}_{ij} = 1\} = \bar{\mathcal{A}}$. Then $(\mathcal{N}, \bar{\mathcal{A}})$ decomposes into a node disjoint collection of simple chains in $G$. The collection of worksequences corresponding to these simple chains is a minimum cost (OBJ2.1) minimum cardinality partition of the set of trips into worksequences.

## How to handle OBJ3

Next we compute the **total working time** for each of the worksequences obtained above, which is the difference between the ending time of the last trip and the starting time of the first trip on the worksequence, expressed in hours. As defined in Section 1, the duration of a worksequence depends on the depot from which the bus for this worksequence is allotted, and it is equal to the total total working time in the worksequence + the driving time from and to the depot. Giving an allowance of one hour for driving from and to the depot, we call worksequences for which the

$$\text{total working time} \geq 12 \text{ hours } (= \text{safety time } -1)$$

as long worksequences, and these are the worksequences most likely to violate the maximum duration constraint.

These long worksequences are of two types. The first type is the single trip worksequences which consist of just one long duration trip, discussed in Section 2. The second type is multitrip worksequences which are long.

Less than 10% customer requests are for long duration trips. The company likes these because they generate higher fees, and they try to assign them with equal frequency among all their drivers. Long duration trips almost always contain nondriving rest periods within its duration during which the driver can take a nap and get refreshed. For this reason single trip long worksequences are never considered a problem.

Since each long duration trip always becomes a single trip worksequence, these long duration trips can be completely left out of the network model discussed above. In the remaining network define the length of arc $(i, j)$ to be $l_{ij} = \bar{t}_j - \bar{t}_i$ = time lapse in hours from the ending time of trip $i$ to the ending time of trip $j$. One way of preventing multitrip long duration worksequences from appearing in the solution is by finding a **feasible simple chain cover** of the nodes in the remaining network, where a feasible simple chain is a simple chain with an upper bound like 12 hours or so for its total length with $l_{ij}$ as arc lengths. However, while a minimal simple chain cover can be found very efficiently, the problem finding a minimal feasible simple chain cover is NP-hard (see [7, 4] for proofs).

OBJ3 is a goal requiring that the percentage of long duration worksequences should be $\leq 50$ as far as possible. We found that in about one third of the days the collection of worksequences obtained do satisfy this goal. On the remaining days the goal is violated, with percentage of long duration worksequences in the collection reaching about 67. On some days when this goal is violated, some of the multitrip long duration worksequences are modified using the following heuristic strategies:

a) A multitrip worksequence of very long duration (over 20 hours) can be broken up into two worksequences of reasonable duration by splitting it in the middle.

b) A worksequence which is long but not very long (between 13 hours and 20 hours), can be made shorter by taking out some trips either at its beginning, or at its end. We then try to include these deleted trips in other worksequences, if possible. Or these deleted trips can be made into worksequences of short working duration by themselves, and assigned to drivers who handled long duration worksequences the previous day.

c) A strategy which has worked very well is the following. On each of these long multitrip worksequences, a longest arc (i.e., an arc $(i, j)$ on this worksequence having maximum $l_{ij}$ value) is deleted from the network $G$, and the algorithm discussed above applied on the remaining network. Almost always this produces a new collection of worksequences that satisfies the goal on the percentage of long worksequences while increasing the number of worksequences only slightly.

Clearly the optimum values of OBJ1, OBJ2.1 obtained in the first run of the algorithm are lower bounds for the minimum values of these respective objective functions in the specified priority order, for partitioning the set of trips considered into worksequences while satisfying the goal in OBJ3. We use these lower bounds to compare the quality of the final solutions obtained.

## Allocating Buses from Sources(depots and outside vendors) to Bus Routes

After finalizing the partition of the set of trips into worksequences, we turn to the problem of assigning buses to these worksequences. These buses can come either from depot 1, 2, or outside vendors. The assignment of buses to worksequences will be carried out so as to minimize

> OBJ2.2 (the cost of empty load drives from the depot to the starting location of the first trip, and from the ending location of last trip to the depot, for all company's own buses used) + the rental cost of buses from outside vendors that are used.

Let

$s$ = number of worksequences

$c_t$, $d_t$ = cost of the empty load drive at the beginning and at the end of the $t^{th}$ worksequence if a bus is assigned to it from depot 1, 2 respectively.

$e_t$ = cost(in \$) of renting a bus for the $t^{th}$ worksequence from an outside vendor.

$N_1$, $N_2$ = number of buses available at depot 1 and 2, respectively.

Since $e_t$ is typically much larger than $c_t$ or $d_t$, the number of buses rented from outsider vendors will be $(s - N_1 - N_2)^+ = \text{Max}\{0,\ s - N_1 - N_2\}$. We have three sources for buses, sources 1, 2, and 3 (these are depot 1, 2, and outside vendors respectively), with availability of buses equal to $N_1$, $N_2$, and $(s - N_1 - N_2)^+$ respectively. Each workequence requires exactly one bus. Clearly the problem of assigning buses to worksequence can be modeled as a $(3\times s)$ transportation problem, (TP1), with the $(3\times s)$ cost matrix whose $t$-th column is $(c_t,\ d_t,\ e_t)^T$ for $t = 1$ to $s$.

## 8 Numerical Results

Results obtained by using our algorithm over requested trip data at the chartered bus company over a 5 day period are shown in Table 1, Table 2, and Table 3. Table 1 shows the number of small group trips, large group trips, long duration trips from original data, and number of small group worksequences,

Table 1: Information about all the trips and the subset of those included among worksequences to which a small size bus is assigned

| Day | $A^*$ | $B^*$ | $C^*$ | $D^*$ | $E^*$ | $F^*$ | $G^*$ | $K^*$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 14 | 50 | 5 | 10 | 4 | 54 | 5 | 49 |
| 2 | 13 | 37 | 5 | 7 | 6 | 43 | 5 | 38 |
| 3 | 12 | 41 | 5 | 9 | 3 | 44 | 4 | 40 |
| 4 | 14 | 46 | 5 | 9 | 5 | 51 | 5 | 46 |
| 5 | 9 | 55 | 4 | 9 | 0 | 55 | 3 | 52 |

$A^*$ : Number of small group trips on day

$B^*$ : Number of large group trips on day

$C^*$ : Number of small group worksequences found, to which a small size bus is assigned

$D^*$ : Number of small group trips included on small group worksequences assigned to small size buses

$E^*$ : Number of small group trips combined with large group trips

$F^*$ : Number of trips for 45 seat buses

$G^*$ : Number of long duration trips among those in $F^*$

$K^*$ : Number of remaining trips after taking out long duration trips

and small group trips included in the worksequences assigned to the available small size buses after applying the algorithm.

Only 5 small size buses are available, but the number of small group worksequences formed usually exceeded 5. In this case, from the collection of those worksequences, the best 5 are selected manually and assigned to the available 5 small group buses, and all the trips on the remaining small group worksequences are combined with the set of large group trips for assignment to a large size bus or a bus from an outside vendor. That is why in Table 2 the number and percentage of long duration worksequences among those assigned to a small size bus is zero on all the five days. Because of this, the final values for OBJ1 and OBJ2.1 for this subset of trips is the same as the lower bound for these respective objective functions for this subset of trips.

Table 3 shows the results on the set of trips (excluding long duration trips) on worksequences to which a larger size bus or an outside vendor bus is assigned. To meet OBJ3, the value of OBJ1 increased by at most 1 over its lower bound, on three of the five days; while OBJ2.1 increases about 9.5% on an average over its lower bound. This shows that final solutions obtained by the approach had objective values quite close to the lower bounds for these objective functions.

The final solutions obtained by the approach also resulted in an average saving of approximately 10% in each of OBJ1 and OBJ2.1 over their values obtained under the manual allocation process that was in use before our study.

Table 2: Results for small group trips among worksequences assigned to a small size bus

| Day | $A^{**}$ | $B^{**}$ | | | $D^{**}$ | | | |
|---|---|---|---|---|---|---|---|---|
| | | OBJ1 | OBJ2.1 | $C^{**}$ | OBJ1 | OBJ2.1 | OBJ2.2 | $E^{**}$ |
| 1 | 14 | 5 | 30,700 | 0 | 5 | 30,700 | 21,700 | 0 |
| 2 | 13 | 5 | 26,500 | 0 | 5 | 26,500 | 24,800 | 0 |
| 3 | 12 | 5 | 23,200 | 0 | 5 | 23,200 | 19,800 | 0 |
| 4 | 14 | 5 | 36,000 | 0 | 5 | 36,000 | 22,000 | 0 |
| 5 | 9 | 4 | 23,000 | 0 | 4 | 23,000 | 18,000 | 0 |

$A^{**}$ : Number of trips for small size buses

$B^{**}$ : First run(lower bound)objective values

$C^{**}$ : Percentage of long duration worksequences in first run

$D^{**}$ : Final objective values

$E^{**}$ : Final percentage of long duration worksequences

Table 3: Results for trips (excluding long duration trips) included on worksequences to which a large size bus or an outside vendor bus is assigned

| Day | $A^{**}$ | $B^{**}$ | | | $D^{**}$ | | | |
|---|---|---|---|---|---|---|---|---|
| | | OBJ1 | OBJ2.1 | $C^{**}$ | OBJ1 | OBJ2.1 | OBJ2.2 | $E^{**}$ |
| 1 | 49 | 22 | 81,000 | 68 | 23 | 95,200 | 77,000 | 48 |
| 2 | 38 | 17 | 84,900 | 53 | 17 | 88,100 | 75,400 | 47 |
| 3 | 40 | 18 | 99,800 | 67 | 19 | 118,000 | 74,800 | 50 |
| 4 | 46 | 21 | 91,500 | 57 | 21 | 96,300 | 84,000 | 48 |
| 5 | 52 | 23 | 95,700 | 61 | 24 | 98,100 | 83,300 | 48 |

$A^{**}$ : Number of trips for 45 seat buses

$B^{**}$ : First run(lower bound)objective values

$C^{**}$ : Percentage of long duration worksequences in first run

$D^{**}$ : Final objective values

$E^{**}$ : Final percentage of long duration worksequences

# 9 Conclusion

In this paper we presented a case study on a chartered bus allocation problem at Arirang in Seoul, South Korea. The approach developed yielded solutions with objective values quite close to the lower bounds for these objective functions. Adopting this approach has been saving the company about 10% of its costs over those incurred using a manual allocation process that was in use before. Data for one sample day is provided in the appendix. This data could be used by faculty members to construct a computational project for their classes.

# Appendix

The following data on 68 trips requiring the large group buses on one day is given as an illustrative example. The various starting/ending locations are numbered serially from 1 to 20. Table 4 is travel time matrix and Table 5 gives the list of trips for that day.

Table 4: Sample travel time (in minutes of driving) for a chartered bus company

| from ⋱ to | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 85 | 45 | 60 | 80 | 25 | 30 | 30 | 55 | 65 | 95 | 110 | 110 | 55 | 35 | 105 | 85 | 65 | 50 | 50 | 85 | 35 |
| 2 | 85 | 0 | 85 | 55 | 35 | 65 | 105 | 105 | 75 | 40 | 40 | 25 | 30 | 30 | 60 | 40 | 55 | 75 | 105 | 40 | 20 | 65 |
| 3 | 45 | 85 | 0 | 35 | 60 | 35 | 35 | 75 | 90 | 85 | 110 | 110 | 100 | 60 | 65 | 80 | 55 | 25 | 25 | 50 | 95 | 25 |
| 4 | 60 | 55 | 35 | 0 | 25 | 40 | 65 | 85 | 85 | 70 | 90 | 80 | 65 | 40 | 60 | 45 | 25 | 20 | 55 | 25 | 70 | 25 |
| 5 | 80 | 35 | 60 | 25 | 0 | 55 | 85 | 100 | 90 | 65 | 70 | 60 | 40 | 35 | 65 | 25 | 20 | 40 | 80 | 30 | 55 | 45 |
| 6 | 25 | 65 | 35 | 40 | 55 | 0 | 40 | 50 | 55 | 55 | 80 | 90 | 90 | 35 | 30 | 80 | 65 | 45 | 50 | 30 | 70 | 15 |
| 7 | 30 | 105 | 35 | 65 | 85 | 40 | 0 | 45 | 85 | 90 | 120 | 130 | 125 | 75 | 60 | 110 | 85 | 60 | 25 | 65 | 110 | 40 |
| 8 | 30 | 105 | 75 | 85 | 100 | 50 | 45 | 0 | 50 | 70 | 105 | 125 | 135 | 75 | 45 | 125 | 110 | 90 | 70 | 70 | 100 | 60 |
| 9 | 55 | 75 | 90 | 85 | 90 | 55 | 85 | 50 | 0 | 35 | 60 | 85 | 105 | 55 | 25 | 110 | 105 | 95 | 105 | 60 | 60 | 65 |
| 10 | 65 | 40 | 85 | 70 | 65 | 55 | 90 | 70 | 35 | 0 | 35 | 55 | 70 | 30 | 30 | 80 | 85 | 85 | 100 | 45 | 30 | 60 |
| 11 | 95 | 40 | 110 | 90 | 70 | 80 | 120 | 105 | 60 | 35 | 0 | 30 | 60 | 50 | 60 | 80 | 90 | 105 | 130 | 65 | 20 | 90 |
| 12 | 110 | 25 | 110 | 80 | 60 | 90 | 130 | 125 | 85 | 55 | 30 | 0 | 30 | 55 | 80 | 55 | 80 | 100 | 130 | 65 | 25 | 90 |
| 13 | 110 | 30 | 100 | 65 | 40 | 90 | 125 | 135 | 105 | 70 | 60 | 30 | 0 | 60 | 90 | 30 | 55 | 80 | 120 | 65 | 50 | 85 |
| 14 | 55 | 30 | 60 | 40 | 35 | 35 | 75 | 75 | 55 | 30 | 50 | 55 | 60 | 0 | 35 | 55 | 55 | 60 | 80 | 15 | 35 | 40 |
| 15 | 35 | 60 | 65 | 60 | 65 | 30 | 60 | 45 | 25 | 30 | 60 | 80 | 90 | 35 | 0 | 90 | 80 | 70 | 80 | 35 | 55 | 40 |
| 16 | 105 | 40 | 80 | 45 | 25 | 80 | 110 | 125 | 110 | 80 | 80 | 55 | 30 | 55 | 90 | 0 | 30 | 60 | 100 | 55 | 60 | 70 |
| 17 | 85 | 55 | 55 | 25 | 20 | 65 | 85 | 110 | 105 | 85 | 90 | 80 | 55 | 55 | 80 | 30 | 0 | 30 | 75 | 45 | 75 | 50 |
| 18 | 65 | 75 | 25 | 20 | 40 | 45 | 60 | 90 | 95 | 85 | 105 | 100 | 80 | 60 | 70 | 60 | 30 | 0 | 45 | 45 | 90 | 30 |
| 19 | 50 | 105 | 25 | 55 | 80 | 50 | 25 | 70 | 105 | 100 | 130 | 130 | 120 | 80 | 80 | 100 | 75 | 45 | 0 | 70 | 115 | 45 |
| 20 | 50 | 40 | 50 | 25 | 30 | 30 | 65 | 70 | 60 | 45 | 65 | 65 | 65 | 15 | 35 | 55 | 45 | 45 | 70 | 0 | 50 | 25 |
| 21 | 85 | 20 | 95 | 70 | 55 | 70 | 110 | 100 | 60 | 30 | 20 | 25 | 50 | 35 | 55 | 60 | 75 | 90 | 115 | 50 | 0 | 70 |
| 22 | 35 | 65 | 25 | 25 | 45 | 15 | 40 | 60 | 65 | 60 | 90 | 90 | 85 | 40 | 40 | 70 | 50 | 30 | 45 | 25 | 70 | 0 |

1 to 20 are the numbers for various locations which are starting/ending locations of trips.

21, 22 are the numbers of the two depots where buses are garaged.

Table 5: Sample requested trips for one day

| Trip | $A^{***}$ | $B^{***}$ | $C^{***}$ | $D^{***}$ | Trip | $A^{***}$ | $B^{***}$ | $C^{***}$ | $D^{***}$ | Trip | $A^{***}$ | $B^{***}$ | $C^{***}$ | $D^{***}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 7 | 5 : 20 | 20 | 6 : 50 | 2 | 20 | 5 : 30 | 3 | 7 : 50 | 3 | 18 | 5 : 40 | 10 | 7 : 40 |
| 4 | 4 | 5 : 40 | 12 | 8 : 10 | 5 | 18 | 6 : 10 | 17 | 8 : 40 | 6 | 11 | 6 : 20 | 16 | 8 : 40 |
| 7 | 20 | 6 : 20 | 10 | 8 : 30 | 8 | 1 | 7 : 00 | 7 | 9 : 00 | 9 | 19 | 7 : 20 | 9 | 20 : 30 |
| 10 | 15 | 7 : 20 | 7 | 8 : 50 | 11 | 20 | 7 : 30 | 1 | 20 : 50 | 12 | 16 | 7 : 40 | 19 | 13 : 00 |
| 13 | 10 | 7 : 50 | 16 | 9 : 00 | 14 | 8 | 7 : 50 | 10 | 23 : 00 | 15 | 8 | 8 : 10 | 8 | 17 : 00 |
| 16 | 20 | 8 : 10 | 15 | 20 : 00 | 17 | 12 | 8 : 10 | 4 | 12 : 00 | 18 | 19 | 8 : 40 | 2 | 18 : 00 |
| 19 | 13 | 8 : 30 | 18 | 12 : 00 | 20 | 14 | 8 : 30 | 7 | 20 : 00 | 21 | 8 | 9 : 00 | 14 | 15 : 00 |
| 22 | 17 | 9 : 00 | 1 | 12 : 00 | 23 | 9 | 9 : 30 | 9 | 18 : 00 | 24 | 20 | 9 : 30 | 20 | 17 : 00 |
| 25 | 17 | 9 : 00 | 2 | 18 : 00 | 26 | 5 | 9 : 30 | 9 | 11 : 30 | 27 | 7 | 9 : 00 | 14 | 18 : 30 |
| 28 | 15 | 10 : 00 | 12 | 20 : 30 | 29 | 3 | 10 : 00 | 11 | 20 : 30 | 30 | 11 | 10 : 30 | 4 | 13 : 30 |
| 31 | 17 | 9 : 00 | 12 | 18 : 30 | 32 | 14 | 11 : 30 | 9 | 16 : 00 | 33 | 6 | 12 : 30 | 9 | 14 : 30 |
| 34 | 17 | 13 : 00 | 8 | 15 : 30 | 35 | 20 | 13 : 00 | 20 | 16 : 50 | 36 | 14 | 13 : 30 | 16 | 18 : 00 |
| 37 | 7 | 13 : 30 | 16 | 15 : 30 | 38 | 17 | 7 : 00 | 18 | 9 : 00 | 39 | 14 | 14 : 30 | 17 | 15 : 30 |
| 40 | 12 | 15 : 30 | 20 | 19 : 00 | 41 | 20 | 16 : 00 | 20 | 20 : 00 | 42 | 9 | 17 : 00 | 6 | 17 : 30 |
| 43 | 20 | 17 : 30 | 16 | 19 : 30 | 44 | 1 | 17 : 30 | 15 | 20 : 30 | 45 | 1 | 17 : 30 | 16 | 19 : 30 |
| 46 | 17 | 18 : 10 | 2 | 20 : 30 | 47 | 11 | 18 : 10 | 11 | 20 : 30 | 48 | 10 | 18 : 20 | 11 | 20 : 30 |
| 49 | 15 | 18 : 20 | 20 | 19 : 50 | 50 | 5 | 18 : 30 | 5 | 20 : 30 | 51 | 20 | 18 : 30 | 2 | 23 : 30 |
| 52 | 16 | 18 : 40 | 3 | 23 : 00 | 53 | 20 | 18 : 30 | 17 | 19 : 50 | 54 | 6 | 18 : 10 | 16 | 20 : 00 |
| 55 | 8 | 19 : 00 | 13 | 23 : 00 | 56 | 8 | 19 : 00 | 15 | 23 : 00 | 57 | 8 | 19 : 00 | 3 | 20 : 00 |
| 58 | 20 | 20 : 00 | 1 | 22 : 30 | 59 | 13 | 19 : 30 | 11 | 20 : 30 | 60 | 20 | 20 : 30 | 7 | 20 : 30 |
| 61 | 10 | 18 : 10 | 15 | 20 : 50 | 62 | 10 | 20 : 00 | 16 | 23 : 00 | 63 | 4 | 20 : 00 | 14 | 22 : 50 |
| 64 | 16 | 18 : 30 | 8 | 22 : 00 | 65 | 20 | 17 : 50 | 11 | 20 : 00 | 66 | 13 | 17 : 30 | 18 | 20 : 00 |
| 67 | 17 | 22 : 00 | 8 | 23 : 00 | 68 | 10 | 20 : 00 | 19 | 20 : 30 | | | | | |

$A^{***}$ : Starting location $B^{***}$ : Starting time $C^{***}$ : Ending location $D^{***}$ : Ending time

# References


[1] Bodin L., Golden B., Assad A., and Ball M. Routing and scheduling of vehicles and crews. *Computers and Operations Research* 1983:63-211.

[2] Carraresi P. and Gallo G. Network models for vehicle and crew scheduling. *European Journal of Operational Research* 1984;16:139-151.

[3] Desrosiers J., Dumas Y., Solomon M.M., Soumis F. Time constrained routing and scheduling. In: Ball et. al.(Eds) *Network Routing*. Amsterdam: North-Holland, 1995. p.35-139.

[4] Fischetti M., Martello S., and Toth P. The fixed job schedule problem with spreadtime constraints". *Operations Research* 1987;35:849-858.

[5] Ford L.R. and Fulkerson D.R. *Flows in Networks*. Princeton University Press: Princeton, New Jersey, USA,1962.

[6] Fulkerson D.R. Note on dilworth's decomposition theorem for partially ordered sets. *Proc. Amer. Math. Soc.* 1956;7:701-702.

[7] Kim W.J., Kabadi S.N., and Murty Katta G. *A chartered bus allocation and scheduling problem*. Technical Report 1999, Department of IOE, University of Michigan, Ann Arbor.

[8] Laporte G. The vehicle routing problem : An overview of exact and approximate algorithms. *European Journal Operational Research* 1992;59:345-358.

[9] Magnanti T. Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks* 1981;11:179-214.

[10] Mole R.H. A survey of local delivery vehicle routing methodology. *Journal of Operational Research Society* 1979;30:245-252.

[11] Murty Katta G. *Network Programming*. Prentice Hall: Englewood Cliffs, New Jersey, USA, 1992.

[12] Solomon M., Baker E., and Shaffer J. Vehicle routing and scheduling problems with time window constraints: Efficient implementations of solution improvement procedure. In Golden B. and Assad A.(Eds.) *Studies in Management Science and Systems* 1988;16.

[13] Vance P., Barnhand C., Johnson E., and Nemhauser G. Airline crew scheduling: a new formulation and decomposition algorithm. *Operations Research* 1997;45(2):188-200.

# Chartered Bus Allocation Problem

Woo J. Kim (Daejin U.), Katta G. Murty, IOE Dept.

U. of Michigan, Ann Arbor, MI-48109-2117, USA

katta_murty@umich.edu

See in Section "Practice Papers Written for Consulting" at URL:

http://www-personal.engin.umich.edu/˜ murty/

Work done for Company Arirang in Seoul, South korea.

They get $n$ (bet. 40 to 90) customer requests for buses with drivers, each day. Each request specifies

Starting location $p_i$

Starting time, $\underline{t}_i$

Ending location $q_i$

Ending time $\bar{t}_i$

Group size

Each request (called **trip**) involves travel with possibly many brief stops, but driver has to be available to group continuously.

Two bus types: 45 seat (20 available), 15 seat (5 available), stationes at two separate depots.

Company can borrow buses from other vendors.

**duration of trip:** = end time - start time

Assume: Start times different

**Trip Number:** Serially in increasing order of start time.

**Worksequence:** Sequence of trips handled by a bus in a day

Necessary cond. for $i_1, \ldots, i_r$ to be a worksequence:

$$\underline{t}_{s+1} - \bar{t}_s \geq \text{driving time from } q_s \text{ to } p_{s+1} \text{ for } s = 1 \text{ to } r-1.$$

**Duration of this worksequence:** (it depends on depot) $= \bar{t}_r - \underline{t}_1 +$ (driving time from depot to $p_1$) + (driving time from $q_r$ to depot)

Driver's pay proportional to duration of worksequence. Drivers prefer worksequences of long durations. But company has **safety limit = 13 hours** for worksequuences.

## Trip Durations

Varies bet. 0.5 hours to 20 hours.

10% of trips are **Long duration trips** ($\geq$ 11 hour duration).

75% of trips have duration $\leq$ 5 hours.

**Goal constraint:** Keep % of worksequences of duration $\geq$ safety limit below 50.

Decisions to be made:

**1.** Partition trips into worksequences for buses.

**2.** Allocate buses from two depots, (and outside vendors if necessary) to worksequences.

Network $G$

Each trip is a node. $\mathcal{N}$ set of nodes.

Include arc $(i, j)$ if bus can handle trip $j$ after completing trip $i$. $\mathcal{A}$ set of arcs. Leave long duration trips as isolated nodes.

$G = (\mathcal{N}, \mathcal{A})$ acyclic. Each worksequence is a chain in $G$, and each chain (including single node chains) in $G$ can be a worksequence.

$c_{ij}$ = Cost of arc $(i, j)$: Arc involves **Empty load drive** from $q_i$ to $p_j$.

$d_{ij}$ = driving time

$w_{ij}$ = waiting time

$\underline{t}_j - \bar{t}_i = d_{ij} + w_{ij}$

$c_{ij} = 40d_{ij} + 30w_{ij}$.

Multiple Objectives

OBJ 1 = no. of worksequences, to be minimized.

OBJ 2 = Total cost of empty load drives and driver waiting times, to be minimized.

OBJ 2 = OBJ 2.1 + OBJ 2.2, where

OBJ 2.1 = cost bet. consecutive trips in worksequnces

OBJ 2.2 = cost from & to depot

OBJ 3 = % of worksequences violating safety limit (to be kept below 50).

## Solution Strategy

OBJ 1 can be minimized by **Dilworth's Minimal Chain Decomposition** of $G$. Each Chain in decomposition is a worksequence in a partition of $\mathcal{N}$ minimizing OBJ 1.

**Fulkerson's Algo. for min chain decomposition**

Define bipartite network $B = \{\mathcal{N}_1, \mathcal{N}_2, \mathcal{A}_1\}$

$\mathcal{N}_1 = \{R_1, \ldots, R_n\}$

$\mathcal{N}_2 = \{C_1, \ldots, C_n\}$

$\mathcal{A}_1 = \{(R_i, C_j) : (i, j) \in \mathcal{A} \quad \text{in } G\}$

$M$ = max. card. matching in $B$, $r = |M|$.

$a$ = set of arcs in $G = \{(i, j) : (R_i, C_j) \in M\}$

$(\mathcal{N}, a)$ breaks into $n - r = n - |M|$ chains, min chain decomposition.

Example: Consider following network.

$$M = \{(R_1, C_2), (R_2, C_5), (R_5, C_7), (R_7, C_{10}), (R_4, C_8), (R_3, C_6), (R_6, C_9)\}.$$

$$a = \{(1, 2), (2, 5), (5, 7), (7, 10), (4, 8), (3, 6), (6, 9)\}.$$

So, can find min OBJ 2.1, min OBJ 1 solution by solving following bipartite matching problem. Define, for $i$, $j = 1$ to $n$

$$x_{ij} = \begin{cases} 1 & \text{if } (R_i,\, C_j) \text{ is in the selected matching,} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} &\textit{Minimize} \quad && Z(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ &\textit{subject to} \quad && \sum_{i=1}^{n} x_{ij} \leq 1 \\ & && \sum_{j=1}^{n} x_{ij} \leq 1 \\ & && \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} = r \\ & && x_{ij} = 0 \text{ if } (R_i, C_j) \text{ is not an arc in } B. \\ & && x_{ij} \in \{0,1\} \text{ if } (R_i, C_j) \text{ is an arc in } B. \end{aligned}$$

## To Handle OBJ 3

In $G$, define $\ell_{ij} = \bar{t}_j - \bar{t}_i$. Then

Duration of any worksequence = length of corresponding chain + driving time from and to depot.

Give allowance of 1 hour for driving time from and to depot, and define

**feasible simple chain** = one of length $\leq 12$ (safety limit -1).

Finding minimum feasible simple chain cover in $G$ is NP-hard.

So, we handled OBJ 3 using some heuristic strategies that worked very well.

### To Minimize OBJ 2.2

A bus for each workstation can come from depot 1, depot 2, or outside vendor. If there are $s$ worksequences in partition selected, problem of allocating buses to worksequences minimizing OBJ 2.2 is a $3 \times s$ transportation problem.