A hybrid optimization algorithm for bus crew scheduling: column generation and local search

Nakorn Indra-Payoong¹, Agachai Sumalee², Kriangsak Vanitchakornpong¹,

¹ Faculty of Logistics, Burapha University, Chonburi, Thailand
² Department of Civil and Structural Engineering, The Hong Kong Polytechnic University, Hong Kong
{nakorn.ii@gmail.com; kriangsv@buu.ac.th; ceasumal@polyu.edu.hk}

ABSTRACT

The paper proposes a hybrid algorithm for solving bus crew scheduling problem (CSP). The CSP involves an assignment of a number of staff to different scheduled bus services. The problem is normally constrained by a number of operational and practical constraints, such as crew preferences, crew satisfaction, and work-shift, etc. These constraints are considered fully in the proposed model. The CSP is typically a large NP-hard problem. The paper handles this by combining column generation (CG) technique with constrained local search method (CLS). The CSP is reformulated on the crew job basis in which a linear relaxation problem (LR) is derived from this reformulation. The CG method iteratively solves the LR with a restricted set of feasible crew jobs (columns). The optimal dual variables from the LR can then be used to form a sub-problem. A solution to this sub-problem with a negative reduced cost is a new column for the LR. The sub-problem, which is NP-hard, with different practical constraints of the CSP will be solved by CLS. The CG will iterate until no additional column with negative reduced cost can be found. The algorithm is tested with the real data set from the bus company in Bangkok. The computational results show that around 20 percent of the existing crews can be reduced. The total operating costs can be decreased significantly while a higher level of crew satisfaction can be achieved.

Keywords: Bus crew scheduling problem, hybrid optimization algorithm, logistics efficiency

1. INTRODUCTION

Crew scheduling problem (CSP) is an important and central part of the logistics management. The CSP involves an assignment of a number of staff to different scheduled bus services. The problem is normally constrained by a number of operational and practical constraints. The main aim of the CSP is to assign the jobs scheduled to different crews in optimal sequences so as to minimize the total costs whilst satisfying several constraints (e.g. job timeslots, work-shift allocation, and job sequencing). Based on an example of a bus line operation in Bangkok, the bus and crew scheduling are carried out separately few months prior to its operations. Prior to bus crew scheduling, job timeslots for each bus line on each day are normally determined as shown in Figure 1.

![Fig. 1: Example of job timeslots](image)

In Fig. 1, there are three shifts for job timeslots: morning, afternoon, and night shifts. Note that different shifts may be assigned with different sign-in/sign-off times for the crews. In this Bangkok example, each crew will have to work for around eight standard working hours per day. The crews can then be assigned different patterns of the shifts over a week or several weeks as illustrated in Fig. 2.
After the crew shift schedule is determined, the bus schedule can be defined separately [1]. Given the bus service schedule, the crew schedules can then be created accordingly.

CSP has been a challenging optimization problem over decades. Several transit crew scheduling systems were developed during 1970 – 1980, such as TRACS, RUCUS, and OPTIBUS [2]. These systems employed heuristic methods to solve the CPS and have been adopted in several countries, such as England, Australia, and Hong Kong. The significant reductions in crew operating costs were reported by the transit companies.

Beasley and Cao [3] proposed a solution algorithm based on the tree search structure to solve CSP. In their algorithm, the subgradient method based on the Lagrangean relaxation is used to increase the efficiency of the lower bound. This algorithm was tested with a general CSP with around 50-500 crew jobs. Peters et al. [4] developed a hybrid GRASP and Branch-and-price method to solve a bus crew scheduling problem in India. Santos and Mateus [5] proposed another hybrid algorithm to solve a bus crew scheduling problem in Brazil. They applied column generation method together with GRASP and genetic algorithm to solve the problem.

The application of artificial intelligence systems was also applied to solve CSP. Ftulis et al. [6] applied the expert system to crew scheduling problems. Operational constraints were formulated as the expert-rules to search for the solution and to efficiently reduce the area of potential search space. Shibhatullah et al. [7] proposed an agent-oriented system to solve dynamic CSP. The dynamic scenario addressed varying traffic conditions and the availability of the crews in case that they do not follow the assigned timetables.

The bus and crew scheduling problems may be integrated and solved simultaneously. However, the advantage of the integrated approach depends largely on characteristics of the problem and current operational procedure of bus companies. Freling et al. [8] compared the sequential and integrated bus and crew scheduling models (VCSP). Valouxis and Housos [9] applied a quick shift heuristic (QS) together with column generation (CG) to solve the VCSP with a case study in Greece. Mesquita and Paias [10] developed the algorithm to tackle VCSP benchmark proposed by Huisman et al. [11]. Rodrigues et al. [12] combined the integer linear program (ILP) and heuristics to solve the VCSP from a case study in Brazil.

The algorithm developed in this paper is relatively similar to [4, 5, 9, 10, 11] which integrates the strengths of the CG and heuristics in handling a large and complex optimization problem. However, each of these algorithms was constructed to suit with different conditions of the case studies. In our algorithm, the constraint-based local search (CLS) approach will be used to generate a feasible set of initial columns and to solve the resulting sub-problems of CG framework instead of the heuristics or meta-heuristics. The application of CLS will enhance the flexibility of the algorithm. Additional side constraints can be considered without modification of the algorithm. This paper, in particular, we consider soft constraints: crew’s duty preference, skilled lines, and duty spread-over which have not addressed in the literature. The paper is structured into further four sections. The next section formulates the CSP. Then, Section 3 explains the solution algorithm developed using the hybrid CG and CLS. Section 4 presents the tests of the algorithm with a real-world case of bus operator in Bangkok. The final section concludes the paper.

2. PROBLEM FORMULATION

Many crew scheduling approaches solve crew pairing and crew rostering sequentially to reduce the complexity of the problem. However, crew pairing and rostering in this paper are considered simultaneously. The main reason is that crew re-scheduling process can be easily maintained when the actual operations deviate from the planned schedules. To facilitate the discussion. Let $W = \{1, ..., w\}$ be the set of jobs, $m$ is the number of crews on a current bus operation, and $M = \{1, ..., m\}$ is defined as the set of crews. Since crew scheduling is planned sequentially after the bus schedules, we could obtain a set of assigned job for crew $i$. $Q_i = \{1, ..., o_i\}$ in ascending order, and Let $Q_i$ be the maximum jobs for crew $i$. Let the denoted parameters in the model are: $c_{ij}$ is a duty operating cost for crew $i$ serving job $j$. $p_{ij}$ is a violation penalty for priority line if crew $i$ is assigned to job $j$. $v_{ij}$ is a violation penalty for priority bus if crew $i$ is assigned to job $j$. $e_{ij}$ is the ordered job $j$ of crew $i$. $t_{ij}$ is a working time of crew $i$ serving job $j$. $\{a_i, b_i\}$ is a working time windows for crew $i$ starting job at time $a$ and finishing job at time $b$. To maintain the duty

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Lot</th>
<th>Job timeslot</th>
<th>Break</th>
<th>Fri</th>
<th>Sat</th>
<th>Sun</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>425102</td>
<td>Crew 1</td>
<td>1</td>
<td>03:25</td>
<td>11:25</td>
<td>Sun</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>432099</td>
<td>Crew 2</td>
<td>2</td>
<td>03:40</td>
<td>11:40</td>
<td>Sat</td>
<td>12</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>439224</td>
<td>Crew 3</td>
<td>3</td>
<td>04:00</td>
<td>12:00</td>
<td>Fri</td>
<td>B</td>
<td>10</td>
<td>7</td>
</tr>
</tbody>
</table>

Fig. 2: Example of shift allocation for crews
spreadover for the assigned crews, $\bar{w}$ is defined as the average jobs for crew, $\bar{w} = \sum_{j \in W} t_j / m'$; where $m'$ is the required number of crews to complete jobs, and $k_i = \sum_{j \in W} t_j x_{ij}$ which is the assigned working hours to crew $i$.

The decision variable in the CSP include the binary variables $x_{ij} \in \{0, 1\}$ in which $x_{ij} = 1$ if crew $i$ serves job $j$ and $x_{ij} = 0$ otherwise. Also, the decision variable $n_i \in \{0, 1\}$ in which $n_i = 1$ if crew $i$ is assigned to job, i.e. $\sum_{j \in W} x_{ij} > 0$ and $n_i = 0$ otherwise. The CSP can be defined mathematically as follow:

$$\begin{align*}
\min \left( \sum_{i \in M} \sum_{j \in W} \left( c_{ij} + p_{ij} + v_{ij} \right) x_{ij} \right) + \sum_{i \in M} n_i + \sum_{i \in M} \left( \bar{w} - k_i \right)^2 \\
\text{s.t.} \quad \sum_{i \in M} x_{ij} = 1 \quad \forall j \in W \\
\left( e_{ij} - e_{i,j-1} \right) - e_{ij} \leq 0 \quad \forall i \in M, j \in O_i \\
a_i - e_{ij} \leq 0 \leq b_j - e_{ij} \quad \forall i \in M, j \in O_i \\
\sum_{j \in W} x_{ij} \leq Q \cdot n_i \quad \forall i \in M \\
x_{ij}, n_i \in \{0, 1\} \quad \forall i \in M, \forall j \in W
end{align*}$$

(1)

The objective function is to minimize the operating costs and the violation penalty for crew preferences. The first constraints ensure that job $j$ is exactly served by one crew. The second constraints require crew can only carry out the next job after completing the previous job. The third constraints ensure that crew executes job after the starting time (a) and after the ending time (b) of the predefined job timeslots, and the fourth constraints require jobs for crew must not exceed the maximum jobs, $n_i$ is served as a slack variable for the crew assignment. The last constraints state that the decision variables in the model take a binary value. The formulation in (1) represents a general form of CSP. However, the structure of the CSP in this formulation involves a large number of variables and constraints. Thus, it is not practical for applying to a large-scale problem. To remedy this, the method of column generation (CG) [13] will be adopted.

To apply the CG, the CSP will be reformulated using the crew based formulation. Let $p \in P_i$ be a work-shift (a sequence of jobs) of crew $i$ in which $P_i$ is a set of feasible work-shifts for crew $i$. Any work-shift in $P_i$ should satisfy all practical constraints as required in (1).

Each work-shift will already be assigned, i.e. $a_{pj}$ is set where $a_{pj} = 1$ if job $j$ is assigned to work-shift $p$ of crew $i$, and $a_{pj} = 0$ otherwise. Let $\lambda_{pj} \in \{0, 1\}$ denotes a dummy variable in which $\lambda_{pj} = 1$ if work-shift $p$ of crew $i$ is adopted, otherwise $\lambda_{pj} = 0$. It is required that for each crew only the maximum of one work-shift can be chosen, i.e. $\sum_{p \in P_i} \lambda_{pj} \leq 1$. Note that crew $i$ may not be used at all, i.e. $\sum_{p \in P_i} \lambda_{pj} = 0$. Also, each job must be served by only one work-shift, i.e. Thus, the CSP can be reformulated as:

\[
\min \sum_{i \in M} \sum_{p \in P_i} \bar{c}_{pi} \hat{\lambda}_{pi}
\]

subject to
\[
\sum_{p \in P_i} \hat{\lambda}_{pi} = 1 \quad \forall i \in M
\]
\[
\sum_{i \in M} \sum_{p \in P_i} a_{pj} \hat{\lambda}_{pi} = 1 \quad \forall j \in W
\]
\[
\hat{\lambda}_{pi} \in \{0,1\} \quad \forall i \in M, \forall p \in P_i
\]

Note that \(\hat{\lambda}_{pi}\) denotes the objective function cost for the work-shift \(p\) of crew \(i\); \(\bar{c}_{pi} = \hat{c}_{pi} + 1 + (w - t_{pi})^2\) where \(\hat{c}_{pi} = \sum_j (c_{pj} + p_{ij} + v_j) x_{ij}\). In this problem, each column of the decision variable corresponds to a feasible work-shift of a crew. The set \(P_i\) can be very large and difficult to enumerate. The next section will present the application CG and CLS to solve (2).

3. SOLUTION APPROACHES

3.1 General framework

The general idea of the solution framework is similar to [4, 5, 9, 10, 11] in which the CG will be used to solve the relaxed problem (RP) of (2). However, the key difference is that an initial set of columns and the sub-problem are handled by constrained local search (CLS) which takes into account all operational constraints and crew preferences of the CSP. The relaxation is only made against the integer requirement of the variables. Each feasible work-shift with assigned jobs is considered as a column in (2). The CLS will be used to find the new columns for the RP. The output from the RP of (2) may not be integer solutions. Thus, the method Branch and Bound (B&B) will be applied to the problem as defined in (2) but formulated with only those columns generated by the CG. Let RP be the relaxed problem of (2) by excluding the integer requirement

\[
(RP) \quad \min \sum_{i \in M} \sum_{p \in P_i} \bar{c}_{pi} \hat{\lambda}_{pi}
\]

subject to
\[
\sum_{i \in M} \sum_{p \in P_i} a_{pj} \hat{\lambda}_{pi} = 1 \quad \forall j \in W
\]
\[
\sum_{p \in P_i} \hat{\lambda}_{pi} = 1 \quad \forall i \in M
\]
\[
0 \leq \hat{\lambda}_{pi} \leq 1 \quad \forall i \in M, \forall p \in P_i
\]

Let \(RP'\) be the restricted version of the RP problem with a subset of possible work-shifts of all crews, i.e.
\[
P' = \bigcup_{i \in M} P_i' \subseteq \bigcup_{i \in M} P_i = P:
\]

\[
(RP') \quad \min \sum_{i \in M} \sum_{p \in P_i'} \bar{c}_{pi} \hat{\lambda}_{pi}
\]

subject to
\[
\sum_{i \in M} \sum_{p \in P_i'} a_{pj} \hat{\lambda}_{pi} = 1 \quad \forall j \in W
\]
\[
\sum_{p \in P_i} \hat{\lambda}_{pi} = 1 \quad \forall i \in M
\]
\[
0 \leq \hat{\lambda}_{pi} \leq 1 \quad \forall i \in M, \forall p \in P_i'
\]

The general framework of the algorithm can be summarized as follows:

Step 1: Generate a set of columns yielding a feasible solution to (2) by using CLS. Let \(P'\) be the set of work-shifts with assigned jobs found by the CLS. Note that using CLS alone can also find a feasible solution to the original CSP, but in the expense of the solution quality.

Step 2: With \(P'\), form RP' and then apply the Simplex method to solve RP'.
Step 3: Define the reduced cost for each work-shift based on the output from Step 2 and then apply CLS to find new work-shifts minimizing the reduced costs. Let \( \hat{P} \) be the set of these feasible work-shifts with assigned jobs.

Step 4: If the reduced costs from \( \hat{P} \) is negative then let \( P' = P' \cup \hat{P} \) and return to Step 2; otherwise proceed to Step 5.

Step 5: If all \( \hat{\lambda}_a \) as output from the Simplex method in Step 2 are integers, then terminate and \( \hat{\lambda}_a \) is the solution to the original CSP. Otherwise, apply the B&B approach to the \( RP' \) with the integral constraint to get integer solution of \( \hat{\lambda}_a \).

From our experiments, when crew preferences become fully considered, often most of \( \hat{\lambda}_a \) found by the Simplex method are integers. Thus, the B&B approach will not take too much time in solving the problem.

3.2 Column generation method

From the linear program, the optimal solution from \( RP' \) will also be the solution of \( RP \) if there is no new column with a negative reduced cost. The reduced cost is defined by the optimal dual variables of the \( RP' \). If there is, then that column should be added into the restricted problem and resolve it again. From (4), the reduced cost can be defined as:

\[
\hat{c}_{pi} = \sum_{j \in W} (c_{ij} - \pi_j)x_{ij} - \pi_i \tag{5}
\]

in which \( \pi_j \) and \( \pi_i \) are the optimal dual variable values corresponding to index \( j \) of the first constraint and index \( pi \) of the second constraint in (4) respectively. To find new columns, one needs to solve (5) for each crew iteratively. Thus the first constraint in the original problem, i.e. the partitioning constraint is relaxed accordingly.

The CLS will be used to solve the sub-problem which is a set-partitioning type formulation. Note that it is possible to solve (4) for all crews in case the constraints of CSP are not defined on individual basis. For instance, the duty spread over when they are considered as hard constraints. The application of CLS for the whole problem will allow more flexibility on the constraint format.

3.3 Constrained local search for generating new columns

The CLS is a type of local search method which is a meta-heuristic optimization method. The algorithm will start with an initial solution and then iteratively moves to neighbour feasible solutions. Thus, the definition of neighbourhood must be defined. The CLS will decide on the move to a neighbour based on some move acceptant criteria. The original CSP and the sub-problem of \( RP' \) can be considered as a set partitioning problem. Figure 3 illustrates the problem format.

**Fig. 3:** Set partitioning format

Figure 3, there are five jobs for this timetable trips with two different service lines: L1 and L2. For instance, bus K11 departs the depot at 8:00 and bus K12 departs at 8:30. c1, c2, c3 and c4 represent crew members. c1 and c2 are dedicated to L1, and c3 and c4 for L2. c1 prefers to work with K11 for some personal reason. This matrix representation will be adopted in the CLS.

Two types of constraints are defined: soft and hard constraints. The hard constraints involve all practical constraints of the CSP which can be defined as in (6) – (8). Note that any other constraints can also be introduced without changing the CLS.

\[
h_{ip} = \max \left( 0, \sum_{j \in W} \left| x_{ij} - 1 \right| \right) \tag{6}
\]

\[
h_{e} = \max \left( 0, \sum_{j \in O} \left( e_{i,j-1} + t_{i,j-1} - e_{ij} \right) \right) \tag{7}
\]
\[ h_{iq} = \max \left( 0, \sum_{j \in W_i} x_{ij} - Q_i \right) \] (8)

Eq. (6) ensures that job is assigned to one crew. Eq. (7) requires crew can only carry out the next job after completing the previous job, and Eq. (8) requires jobs for crew must not exceed the maximum jobs is the violation penalised when a crew exceeds the maximum jobs.

Let \( H = h_{iu} + h_{ic} + h_{ib} \) be the total violation function of the hard constraints. \( H \) is required to be 0 due to the definition of the hard constraint. The soft constraint is associated with the objective function of the sub-problem:

\[ s_{ia} = \max \left( 0, n_i \right) \] (9)

\[ s_{ic} = \max \left( 0, \sum_{j \in W_i} \hat{c}_{ij} x_{ij} \right) \] (10)

\[ s_{ib} = \max \left( 0, t_{io} - b_j \right) \] (11)

\[ s_{ia} = \max \left( 0, (\bar{w} - k_i)^2 \right) \] (12)

\[ s_{ip} = \max \left( 0, \sum_{j \in W_i} p_{ij} x_{ij} \right) \] (13)

\[ s_{iv} = \max \left( 0, \sum_{j \in W_i} v_{ij} x_{ij} \right) \] (14)

\[ s_{iz} = \max \left( 0, \sum_{j \in W_i} z_{ij} x_{ij} \right) \] (15)

Eq. (9), \( s_{ia} = 1 \) if crew \( i \) is required and \( s_{ia} = 0 \) otherwise. Eq. (10) – (11) are the duty and overtime cost respectively. Eq. (12) maintains the duty spread-over for crews where the number of crews assigned to jobs in a current solution, \( m' = \sum_{i=1}^{\lambda_m} \). Eq. (13) – (14) assigns crew to line and specific bus with different priority levels. Eq. (15) is the violation weight for crew assignment. This facilitates the column generation-based approach to obtain the convergence of the CSP solution where a number of optimal points exist. Note that Eq. (15) will be discarded from the solution once the optimized crew schedule has been achieved. Let \( S = s_{iu} + s_{ic} + s_{ia} + s_{ip} + s_{iv} + s_{iz} \) which represents the total soft constraint level.

**Sequencing feasible jobs**

Since crew and bus scheduling is independent, and is planned sequentially, we can pre-assign the starting time of crew jobs in ascending order, i.e. \( e'_{j+1} < e'_j \), \( \forall j \in W \) where \( e'_j \) is the starting time of job \( j \)

**Constrained local search**

The CLS will move from the current solution to its neighbours to reduce \( V = S + H \). The neighbourhood search operators adopted in the CLS is a simple trial flip operation. The CLS will ensure the satisfaction of the hard constraints as defined in (6) – (8) and minimize the total violation penalty of soft constraints as defined in (9) - (15) by its search operators. The procedure of CLS can be summarized as the following steps:

**Step 1:** Initialization. Randomly assign crews to jobs correspond to complete assignment \( A \) of 0 or 1 to all decision variables, in which some hard constraints in the model can be violated.

**Step 2:** Constraint selection. Randomly select a violated hard constraint, e.g. the assigned jobs for crew that exceed the predefined maximum jobs. More violated constraints can also be applied to diversify the search.

**Step 3:** Variable selection. Having selected a violated constraint, the CLS randomly selects one variable in that constraint and another variable, either from the violated constraint or from the search space. Then, the trial
flips will be performed, i.e. changing the current value of the variable to its complementary binary value, resulting the total constraint violation \( V' \) as illustrated in Fig. 4.

![Fig. 4: Trial flip operations](image)

Step 4: Move acceptance. Choose the best \( V' \) amongst the trial flip operations. The CLS will move to a new solution \( A' \rightarrow A \) whether \( V' < V \) or not. This will help the CLS to avoid local optima.

Step 5: Stopping criteria: Terminate when the stopping criteria, e.g. the specified number of iterations, are met.

The procedure can be readily modified to make more use of and several constraint and variable selection rules and to improve the performance of the solving algorithm, e.g. some constraint propagation techniques. On a hybrid CG framework, CLS is mainly used to generate a good initial set of feasible columns to (4) as the search space is fully explored by its diversification strategy in CLS, e.g. a complete assignment of all variables in the model is performed. Also, CLS is used to find the solution for sub-problem. The sub-problem is represented by an individual crew (column) together with a dummy column. With the dummy, we need not to modify a trial swap process or any changes in structure of the algorithm, and the violation for the dummy column is ignored during the search.

### 4. COMPUTATIONAL EXPERIMENT

We test the proposed model and algorithm using the data from the Bangkok mass transit authority (BMTA), Thailand. There are currently 3,535 buses with around 7,000 crews (as driver) operating 108 routes (lines) clustered into 8 zones in the Bangkok city. In general, there are 3 - 4 depots for each zone, and each depot operates 5-10 bus lines. The bus and crew schedules are currently determined by the BMTA planner sequentially with no computer aided tool every few months prior to its operations. In the crew scheduling model, the soft constraints can be categorized as: i) pre-specified constraint, ii) crew preference, and iii) schedule cost, as shown in Table 1.

<table>
<thead>
<tr>
<th>No.</th>
<th>Soft constraints</th>
<th>Violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>Crew pair</td>
<td>0</td>
</tr>
<tr>
<td>A-2</td>
<td>#Break</td>
<td>0</td>
</tr>
<tr>
<td>A-3</td>
<td>Break day</td>
<td>0</td>
</tr>
<tr>
<td>A-4</td>
<td>Shift type</td>
<td>0</td>
</tr>
<tr>
<td>A-5</td>
<td>Sing-in/Sign-off time</td>
<td>0</td>
</tr>
<tr>
<td>B-1</td>
<td>Priority bus</td>
<td>5</td>
</tr>
<tr>
<td>B-2</td>
<td>Priority line (4 levels)</td>
<td>0,5,15,30</td>
</tr>
<tr>
<td>B-3</td>
<td>Duty spreadover</td>
<td>1</td>
</tr>
<tr>
<td>C-1</td>
<td>#Crew</td>
<td>500</td>
</tr>
<tr>
<td>C-2</td>
<td>Duty operating cost (DOC)/min</td>
<td>1.05</td>
</tr>
<tr>
<td>C-3</td>
<td>Overtime (OT)/min</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: Soft violation scheme

From Table 1, the soft constraints in the pre-specified type will not definitely be violated. This is simply because crew pair, #break, break day, shift type, and sign-in/sign-off time are pre-assigned and not included as decision variables in the model. This represents the actual operational practice and limits the size of the search space. The soft constraints for
crew preference include the priority bus, priority line, and duty spread-over. The violation for the priority line is determined as four levels: 0, 5, 15, and 30 respectively, i.e. the zero violation represents the highest crew preference. The constraint violations for the schedule cost include crew cost, duty operating cost (DOC), and overtime cost (OT), which can be specified directly from the operational costs spent. The total soft violation can be used as a measure for the efficiency of the optimized crew schedules, and BMTA can test on different soft violation schemes in order to evaluate the efficiency of the schedules.

With a single skill line policy, we compare the results obtained from the proposed model with the current schedules which are defined manually by the experts in the filed. In order to assess the performance of the proposed solution framework, we further compare the results obtained by hybrid algorithm and those obtained by CLS alone. Both algorithms are tested for 10 different runs with the CPU time given in Seconds. To make a fair comparison, we stop the hybrid algorithm as the amount of time CLS terminated, i.e. 1,500 no improving iteration is found. Cost represents the total schedule cost and Time to best is the time that CLS finds the best solution. The test results are given in Table 2.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>75</td>
<td>48</td>
<td>25.4</td>
<td>19,834</td>
<td>19,696</td>
<td>14</td>
<td>24</td>
<td>19,182</td>
<td>19,155</td>
<td>14</td>
</tr>
<tr>
<td>P2</td>
<td>75</td>
<td>48</td>
<td>23</td>
<td>16,041</td>
<td>15,947</td>
<td>11</td>
<td>22</td>
<td>15,520</td>
<td>15,520</td>
<td>11</td>
</tr>
<tr>
<td>P3</td>
<td>75</td>
<td>48</td>
<td>24</td>
<td>17,023</td>
<td>16,982</td>
<td>15</td>
<td>23</td>
<td>16,471</td>
<td>16,442</td>
<td>15</td>
</tr>
<tr>
<td>P4</td>
<td>75</td>
<td>48</td>
<td>26</td>
<td>21,005</td>
<td>20,999</td>
<td>10</td>
<td>24</td>
<td>19,383</td>
<td>19,322</td>
<td>10</td>
</tr>
<tr>
<td>P5</td>
<td>75</td>
<td>48</td>
<td>19</td>
<td>13,236</td>
<td>13,221</td>
<td>9</td>
<td>18</td>
<td>12,950</td>
<td>12,950</td>
<td>9</td>
</tr>
<tr>
<td>P6</td>
<td>72</td>
<td>36</td>
<td>25</td>
<td>19,885</td>
<td>19,885</td>
<td>10</td>
<td>24</td>
<td>19,416</td>
<td>19,410</td>
<td>10</td>
</tr>
<tr>
<td>P7</td>
<td>72</td>
<td>36</td>
<td>24</td>
<td>19,530</td>
<td>19,422</td>
<td>8</td>
<td>24</td>
<td>19,285</td>
<td>19,285</td>
<td>8</td>
</tr>
<tr>
<td>P8</td>
<td>72</td>
<td>36</td>
<td>25.5</td>
<td>21,492</td>
<td>20,947</td>
<td>18</td>
<td>24</td>
<td>19,756</td>
<td>19,756</td>
<td>18</td>
</tr>
<tr>
<td>P9</td>
<td>72</td>
<td>36</td>
<td>24</td>
<td>19,329</td>
<td>19,211</td>
<td>15</td>
<td>24</td>
<td>19,014</td>
<td>18,945</td>
<td>15</td>
</tr>
<tr>
<td>P10</td>
<td>72</td>
<td>36</td>
<td>26</td>
<td>22,032</td>
<td>22,004</td>
<td>20</td>
<td>25</td>
<td>21,498</td>
<td>21,329</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 2: Single skill policy

Table 2, comparing the results obtained from the optimized schedules (by CLS and Hybrid) with the manual plan, we can cut off at least (xx%) unnecessary crews. Note also that even 10% of crews is included additional as redundancy crew, the reduction in terms of the crew number is still significant.

From the computational point of view, the hybrid method slightly outperforms CLS alone in terms of the schedule cost, but carry a computational effort with a more complexity of the algorithm structure. In all cases, CLS performs satisfactorily which find the best schedule cost in early iterations, but the solution improvement is hardly found during further iterations as shown in Fig. 5. Some improvement by the hybrid method can steadily be found during the end of the search but not significant. If testing several soft violation schemes or quick optimized crew schedule is required, CLS may be a preferred option.

In this paper, crews with multi-skilled line scenarios are proposed to reduce the operating costs by extending the capacity of the current resources. In particular, our interest is the allocation of crews sequentially followed the mixed
bus line operations, which is one of the potential strategies to increase the crew capacity. The test results from the hybrid algorithm are shown in Table 3.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Trips</th>
<th>#Crew</th>
<th>Preference Cost</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single 1</td>
<td>259</td>
<td>88</td>
<td>819</td>
<td>0</td>
</tr>
<tr>
<td>2-Line</td>
<td>250</td>
<td>84</td>
<td>184</td>
<td>1609</td>
</tr>
<tr>
<td>3-Line</td>
<td>250</td>
<td>82</td>
<td>692</td>
<td>1495</td>
</tr>
<tr>
<td>Single 2</td>
<td>250</td>
<td>85</td>
<td>931</td>
<td>0</td>
</tr>
<tr>
<td>2-Line</td>
<td>250</td>
<td>206</td>
<td>135</td>
<td>1294</td>
</tr>
<tr>
<td>3-Line</td>
<td>250</td>
<td>80</td>
<td>203</td>
<td>403</td>
</tr>
<tr>
<td>Single 3</td>
<td>254</td>
<td>82</td>
<td>692</td>
<td>0</td>
</tr>
<tr>
<td>2-Line</td>
<td>254</td>
<td>81</td>
<td>793</td>
<td>293</td>
</tr>
<tr>
<td>3-Line</td>
<td>254</td>
<td>80</td>
<td>823</td>
<td>403</td>
</tr>
</tbody>
</table>

Table 3: Multi-skilled policy

Form Table 3, it is also obvious that the computerized crew schedules with a single-skilled policy can cut-off the significant number of assigned crews compared with the current method. However, there is only a small improvement in terms of crews and the total violation for crew schedules with multi-skilled policy. This is because the number of crew jobs is minimized and fixed as priori during the bus scheduling process, particular in a mixed bus line operations. The soft violations for crew preferences independently have no meaning depending on the soft violation schemes as set in Table 1, but they become useful when compared in the same preference type across test runs. For the priority bus constraint (preference 1), its violation tends to increase for crew schedules with multi-skilled policy; noted also that this increased trend is not always true because in practice they may exist only some crews state or are given a priority to specific bus types, and the degree of violation is implicitly tradeoff with other crews’ preferences.

The violation penalty for priority line (preference 2) is increased when more skill-lines are scheduled as some crews are assigned to jobs that violate their preferences. In contrast, the violation penalty for duty spread-over (preference 3) is reduced largely for the schedules with multi-skilled lines as crews are assigned crossing between lines. Therefore, the effective crew schedules may depend largely on the policy of the company and how they tradeoff between multi business criteria. For instance, if crew with a highly driving skill is seriously concerned in a congested urban area, then the multi-skilled policy shall not be applied.

We also compare the search convergence of the hybrid algorithm and CLS to P12 which become relatively large-sized when crews with multi skills are applied. This is shown in Fig. 6.

Fig 6: Search convergence for P12

Fig. 6 shows that the hybrid algorithm can converge to the near-optimal solution steadily quickly within a reasonable time whilst the CLS alone can move down for a better solution in the early stage and then are trapped for a number of
iterations with a very slow improvement. This is a case for CLS when the search space become large and without any special tunings from the problem specific knowledge. The hybrid method takes advantage of the optimal dual variable prices from Linear relaxation that can intensify the search iteratively adding more fruitful columns to the subproblem, and CLS in a hybrid framework can thus effectively diversify the search within the restricted area.

5. CONCLUSIONS

The paper proposed a hybrid solution framework combining the column generation technique (CG) and constrained local search (CLS) algorithm for solving the bus crew scheduling problem (CSP). The advantage of the hybrid algorithm developed is its non-domain specific and enhances the stability and robustness of the algorithm. The computational experiments are performed to evaluate the performance of the proposed method compared with the current operations. The results based on the BMTA data showed that the number of crews and the total operating costs can be saved significantly by the computerized schedules with a reasonable computational run-time. In addition, in all test cases, the schedule costs from the hybrid algorithm are better than those with CLS method alone. However, incorporated with the local search nature, the hybrid algorithm cannot guarantee the optimality of the solution found. Note that the optimized crew schedules may not directly reflect the company policy on its actual operations. Thus, it is necessary that the schedule should be tested with several soft violation schemes so as to select the most appropriate crew schedule. Future research will look into the dynamic reassignment for unpredictability of traffic and crew availability in which real-time information is applied to generate more reliable and robust bus and crew schedules in day-to-day operations.

ACKNOWLEDGMENTS

This research has been supported in part by a grant from the National Electronics and Computer Technology Center (NECTEC), Grant No. 09/2550 NT-B-22-IT-26-50-09). The authors are also grateful to the BMTA for help and support.

REFERENCES