Heuristic Algorithm for a Military Training Timetabling Problem

Si-Young Leea, Yeong-Dae Kimb,*, Ho-Joo Leec

aProgram Management Agency, Defense Acquisition Program Administration, Korea
bDepartment of Industrial Engineering, Korea Advanced Institute of Science and Technology, Korea
cSystem Analysis Center, Training and Doctrine Command, Republic of Korea Army, Korea

Accepted 27 August 2008

Abstract

This paper focuses on a military training timetabling problem (MTTP) that occurs in the Korea army. The MTTP can be considered a generalized version of the professor-lecturer model composed of classes, groups of classes, professors for group lectures and lecturers for class lectures. Unlike the typical professor-lecturer model, we need to consider additional constraints to reflect real situations, such as those for lunch time, duration of each lecture (2 hours and 4 hours), and set-up time required between the lectures if places (indoor and outdoor) for two consecutive lectures are different. We suggest a heuristic algorithm for the MTTP with the objective of minimizing the makespan and the number of setups. This algorithm employs solution methods for the edge-coloring problem, in which solutions are found by edge-ordering rules. Results of computational experiments show that the suggested algorithm gives good schedules in a reasonably short time.

Keywords: Military training, timetabling, edge coloring, bipartite graph

1. Introduction

Every year, a large number of recruits are trained by the army in Korea and over half of them are trained at the Korea Army Training Center (KATC). At KATC, a battalion, a basic unit for training, is composed of 800 to 900 recruits. All recruits in the same battalion follow exactly the same course of training. In other words, they are all trained with the same subjects at the same period of time. However, the current educational system at KATC has problems in that loads of instructors are not well balanced and efficiency of training is not high enough. Therefore, they plan to restructure the educational system of KATC in order to improve the quality and efficiency of training.

At KATC, a battalion is divided into 16 classes (platoons) and four classes constitute a group (company). Lectures are given inside a building or outside, and the length of a lecture may be either 1 period (2 hours) or 2 periods (4 hours). There are four periods (unit times) for the lectures in a day and there is a lunch time between the second and third periods. Lectures are given to groups or classes. A group-lecture is given to a group by a company leader (professor), while a class-lecture is given to a class by a platoon leader (lecturer). Lectures that each lecturer or professor should give and places for the lectures (indoor and outdoor) are given. Between indoor lectures and outdoor lectures, there should be a preparation (set-up) time required for the lectures because recruits have to move to the places for lectures and prepare teaching material and/or tools for the classes.

In this paper, we consider a military timetabling problem (MTTP) with the objectives of

* Corresponding author. E-mail: ydkim@kaist.ac.kr
minimizing makespan, i.e., the maximum completion time of the lectures, and minimizing the total number of set-ups. There have been various research results on the timetabling problems (de Werra, 1970; 1985; Costa, 1994; Fahrion and Dollansky, 1992). A large portion of the research is based on methodologies for graph problems since timetabling problems can be converted to graph problems as de Werra (1997) gives several examples. Especially, the problem is often modeled as an edge-coloring problem of a bipartite graph (de Werra, 1985; 1996; Asratian and de Werra, 2002). A survey on research on timetabling problems and recently developed approaches for the problems is also given in Burke and Petrovic (2002).

In this study, the MTTP is modeled as an edge-coloring problem. Based on the graph theory related to the edge-coloring problem (Schrijver, 1982; Asratian and Kamalian, 1994; Jain and Werth, 1995; Kapoor and Rizzi, 2000; Caragiannis et al., 2002; Makino et al., 2002), we develop a heuristic algorithm for the MTTP. In this algorithm, we find a special type of matchings using edge-ordering rules, and then we color the matchings with colors that are associated with time units.

This paper is organized as follows. In the next section, we discuss relationships between the timetabling problem and the edge-coloring problem of bipartite graphs. Also, we describe the MTTP considered in this study in more detail by presenting special characteristics (constraints) that should be considered in real situations. In Section 3, we present a heuristic algorithm for the MTTP based on solution methods for matching problems. Performance of the algorithm is evaluated through a series of computational experiment, and results are reported in Section 4. Finally, Section 5 concludes the paper with a short summary.

2. Timetables and edge coloring on bipartite graphs

As mentioned earlier, there are similarities between timetabling problems and edge coloring problems. In the following, we describe the relationships between timetables and proper edge colorings of bipartite graphs by using the popular timetabling models.

2.1 Class-teacher model

A **class-teacher model** is composed of classes, teachers and an $n \times m$ requirement matrix, $B = (b_{ij})$. In this model, there are $n$ classes ($C_1, C_2, ..., C_n$) that should be taken care of by $m$ teachers ($T_1, T_2, ..., T_m$). The element $b_{ij}$ in the requirement matrix specifies the number of lecture units that teacher $T_j$ must give to class $C_i$. A timetable is represented by $S = (s_{ij})$, an $n \times H$ matrix composed of classes and periods, where $H$ is the number of periods in the planning horizon. Each entry of the solution matrix, $S$, specifies the teacher who should give a lecture to class $i$ in time period $t$. Therefore, $S$ should satisfy: each entry is either one of the members of the set $\{T_1, T_2, ..., T_m\}$ or null; and the symbol $T_j$ appears exactly $b_{ij}$ times in the $i$th row. A teacher can give a lecture to only one class in a period, and each class can be taught by only one teacher.

The class-teacher model can be transformed to an edge coloring problem. Let $V$ denote the node set of a bipartite graph, $G$, that consists of class nodes $V_1$ and teacher nodes $V_2$. In this graph, an edge connecting node $C_i$ and node $T_j$ represents an assignment of teacher $T_j$ to one period of lecture given to class $C_i$. Therefore, we insert $b_{ij}$ parallel edges, edges with the same pair of incident nodes, i.e., node $C_i$ (class) and node $T_j$ (teacher). A proper edge coloring of a graph $G$ with colors $\alpha_1, \alpha_2, ..., \alpha_H$ is an assignment of colors to the edges of $G$ in which no pair of adjacent edges receives the same color. One easy way for proper edge coloring is to decompose the edges into matchings and color the matchings with different colors. Note that each of these colors corresponds to a time slot in the timetable. These relationships are illustrated in Figure 1, in which a matching given in the left figure represents an assignment given in the matrix.
2.2 Professor-lecturer model

The MTTP is more closely related with the professor-lecturer model, in which there are four types of elements: classes, groups, lecturers and professors. In the professor-lecturer model, there are $n$ classes ($C_1, C_2, ..., C_n$) that should be taken care of by $m$ lecturers ($L_1, L_2, ..., L_m$). The set of classes is partitioned into $p$ disjoint groups ($G_1, G_2, ..., G_p$) that should be taught by $q$ professors ($P_1, P_2, ..., P_q$). There is an $n \times m$ requirement matrix, $B = (b_{ij})$, which shows the number of lecture units that teacher $L_j$ must give to class $C_i$. Also, there is a $p \times q$ requirement matrix, $A = (a_{kl})$, which shows the number of lecture units that professor $P_l$ must give to group $G_k$.

Consider a bipartite graph with node sets $(V_1, V_2)$, where $V_1 = \{C_1, C_2, ..., C_n, G_1, G_2, ..., G_p\}$ and $V_2 = \{L_1, L_2, ..., L_m, P_1, P_2, ..., P_q\}$. Here, nodes $C_i$ and $L_j$ are joined by $b_{ij}$ parallel edges and nodes $G_k$ and $P_l$ are joined by $a_{kl}$ parallel edges. There is no edge between $C_i$ and $P_l$ or between $G_k$ and $L_j$ in the professor-lecturer model. Then, $S$ should satisfy: each entry is either one of the elements of the set $\{L_1, L_2, ..., L_m, P_1, P_2, ..., P_q\}$ or null; and symbol $L_j$ appears exactly $b_{ij}$ times in the $i$th row; and $P_l$ occurs $a_{kl}$ times in every row associated with a class that is included in group $G_k$. A lecturer can teach only one class in a period, and each class can be taught by only one lecturer. Similarly, a professor can teach only one group of classes in a period, and each group can be taught by only one professor.

Unlike the class-teacher model, an infeasible solution may be obtained in the professor-lecturer model if the solution procedure for the class-teacher model is used in this model. Therefore, we need a special type of matching to obtain a feasible solution in this model. For a matching to be feasible in the professor-lecturer model, the following constraints, to be called the group constraints, should be satisfied in addition to those for a typical matching for the class-teacher model.

(a) If an edge incident to a class is included in a matching, any edge incident to the group containing the class cannot be included in the matching.

(b) If an edge incident to a group is included in a matching, any edge incident to classes that are elements of the group cannot be included in the matching.

These relationships are illustrated in figure 2, in which two matchings colored with different colors given in the left figure represent an assignment given in the matrix.
2.3 The military training timetabling problem

The military training timetabling problem (MTTP) considered in this study is the problem of assigning lecturers and professors to the classes and groups for the bi-criterion objective of minimizing the makespan and the number of setups needed. The MTTP is similar to the professor-lecturer model in that a lecturer (professor) gives a lecture to only one class (group) in each period, and a class (group) should be lectured by one lecturer (professor) in each period.

However, in MTTP, there are a few additional constraints we need to consider to reflect real situations. One is the lecture continuity constraint that comes from the lengths of the lectures. There are two types of lectures in the MTTP, ones requiring two periods and ones requiring one period. Both the lecturers and professors can give either one-period lectures or two-period lectures, but they can be assigned to only one type of lecture. If a two-period lecture is started in a period, it should be continued in the next period. Another constraint is related to the start time of the two-period lectures. In the training center considered in this study, a day consists of 4 periods and there is a lunch time between the second and third periods. Therefore, we have to start two-period lectures at the first or third period. Also, every group has the same number of classes, since groups represent military troops with the same number of subordinate troops. In addition, a lecturer (professor) must give the same number of lectures to the classes (groups) because all recruits in a battalion have to follow the same training course. These characteristics are described in Figure 3.

Figure 2. Relationship of a matching and a solution of timetabling problem in a professor-lecturer model.

Figure 3. Relationship of a matching and a solution of timetabling problem in the MTTP.
3. Heuristic algorithm

In this section we present a heuristic algorithm for the military training timetabling problem (MTTP). As discussed earlier, since this problem is a generalized version of the professor-lecturer model, we can convert this timetabling problem to an edge-coloring problem of a bipartite graph. Our algorithm for the MTTP is based on methods for decomposition of the bipartite graph into matchings. In the algorithm, a solution can be obtained if all matchings are colored with different colors. However, we need a special type of matchings to obtain a feasible solution of the MTTP, since there are additional constraints in the MTTP.

In the heuristic algorithm, edges are ordered with a certain ordering rule, and then a matching that satisfies the above additional constraints (the lecture continuity constraint and the start time constraint for two-period lectures) is selected through a search of the edges in that order, and a color is assigned to the selected matching. This procedure is repeated until every edge in the graph is colored. First, we describe how the edges are ordered in the heuristic algorithm.

3.1 Edge ordering

In the heuristic algorithm, we consider the loads of the lecturers and professors, i.e. the number of lectures given by a lecturer or a professor. In other words, we consider the degree of (the number of edges incident to) the nodes associated with the lecturers and professors. If a lecturer node has many incident edges in the graph, the lecturer has many lectures to give. Also, if a class node has many incident edges, the class has many lectures to get from lecturers. The degree of a node is used for ordering the edges and determining how early each edge should be considered as a candidate for a matching.

In this study, as the edge ordering rule, we use the highest combined degree first rule (HCDF) described by Jain and Werth (1995). In this rule, edges are sorted in a non-increasing order of the sum of the degrees of their endpoints, i.e. two nodes connected to each edge. In other words, a higher priority is given to an edge whose corresponding class and teacher have more remaining lectures. However, in this method, class nodes and group nodes are not distinguished from each other. Therefore, we suggest another ordering rule, named modified highest combined degree first rule (MHCDF). In MHCDF, (work)load of a class is computed as the sum of the load of the class and the load of the group that includes the class. There may be three variations in MHCDF, since one can use the maximum, average or minimum value among the loads of class nodes included in the group to compute the load of a group node.

In the following, we describe the ordering rules in more detail. In the description, the following notation is used. Note that the notation is slightly different from the earlier one.

\[
\begin{align*}
C_e & = \text{class node incident to edge } e \\
L_e & = \text{lecturer node incident to edge } e \\
G_e & = \text{group node incident to edge } e \\
P_e & = \text{professor node incident to edge } e \\
W(e) & = \text{(work)load of edge } e \\
W'(e) & = \text{load of edge } e \text{ in which weight factors are considered} \\
I(x) & = \text{number of edges incident to node } x \\
N(G_e) & = \text{number of classes included in group node } G_e. \\
F(G_e) & = \text{set of class nodes included in group } G_e \\
g(C_e) & = \text{group node associated with the node in which the class associated with } C_e \text{ is included}
\end{align*}
\]
$T^I$ = group of teachers giving indoor lectures
$T^O$ = group of teachers giving outdoor lectures
$w_1$ = weight related to the groups (0 ≤ $w_1$ ≤ 1)
$w_2$ = weight related to the lecture places (0 ≤ $w_2$ ≤ 1)

Four ordering rules developed in this study are given below with the functions used to compute the load of an edge, $e$.

**HCDF**

$$W(e) = I(C_e) + I(L_e)$$  
if edge $e$ is incident to class nodes.

$$W(e) = I(G_e) + I(P_e)$$  
if edge $e$ is incident to group nodes.

**MHCDF-max**

$$W(e) = I(C_e) + I\{g(C_e)\} + I(L_e)$$  
if edge $e$ is incident to class nodes.

$$W(e) = I(G_e) + \max[I(C_i) | C_i \in F(G_e)] + I(P_e)$$  
if edge $e$ is incident to group nodes.

**MHCDF-avg**

$$W(e) = I(C_e) + I\{g(C_e)\} + I(L_e)$$  
if edge $e$ is incident to class nodes.

$$W(e) = I(G_e) + \sum_{C_i \in F(G_e)} \frac{I(C_i)}{N(G_e)} + I(P_e)$$  
if edge $e$ is incident to group nodes.

**MHCDF-min**

$$W(e) = I(C_e) + I\{g(C_e)\} + I(L_e)$$  
if edge $e$ is incident to class nodes.

$$W(e) = I(G_e) + \min[I(C_i) | C_i \in F(G_e)] + I(P_e)$$  
if edge $e$ is incident to group nodes.

In addition to the above four rules, we can make variations of the above rules by multiplying certain weights. These weights are introduced for giving higher priorities to group nodes in the rules. If an edge incident to a group node is considered as a candidate for matching at a later stage of an algorithm, it can hardly be selected because more constraints should be satisfied for it to be selected than for an edge incident to a class node. Therefore, we may obtain better solutions for the MTTP, if we consider an edge incident to a group node earlier. In addition, these weights are introduced for giving higher priorities to edges that do not incur setups. Note that if selection of an edge does not incur a setup, the edge may be considered a more desirable candidate for matching. Therefore, using such weights may result in a better timetable with a shorter makespan and less setups. We introduce two types of weights. One type of weights ($w_1$) is applied to edges incident to group nodes, and the other type of weights ($w_2$) is related to setups. A better solution can be obtained if an edge is incident to a node related to a teacher that teaches in the period, period $t$, at the same place (indoor or outdoor) as that for period $t-1$, since it may result in less setups.

If an edge, say edge $e$, is incident to a group node, we let the weighted load of edge $e$ as $W'(e) = (1 + w_1) W(e)$, where 0 ≤ $w_1$ ≤ 1. In other words, the load of an edge incident to a group node is set to be higher than that of an edge incident to a class node. In addition, if edge $e$ is incident to a node corresponding to a teacher that gives a lecture at the same place as the one for the immediately preceding lecture, we let $W'(e) = (1 + w_2) W(e)$, where 0 ≤ $w_2$ ≤ 1. Note that, however, if a setup is needed before the first or the third period of a day, it can be ignored because the setup can be performed before the day or during the lunch time. Therefore, the above is applied to the second and the fourth periods only. In this paper, we select the values of the weights from six candidate values (0, 0.2, 0.4, 0.6, 0.8 and 1) after testing them on a number of test problems.

The procedure for ordering edges is summarized below. This procedure is applied to each
period independently.

Procedure 1 (Edge ordering for period \(t\))

Step 1. If \(t\) is an odd number, go to Step 2. Otherwise, go to Step 3.

Step 2. Compute weighted loads of the edges, \(W'(e)\), as follows: let \(L'(e)\leftarrow W(e)\) for edges incident to class nodes; and let \(W'(e)\leftarrow (1 + w_1)W(e)\) for edges incident to group nodes. Go to Step 4.

Step 3. Compute weighted loads of the edges, \(L'(e)\), as follows: let \(W'(e)\leftarrow W(e)\) for edges incident to class nodes and teacher nodes with a different lecture place from the one for period \(t-1\); let \(W'(e)\leftarrow (1 + w_2)W(e)\) for edges incident to class nodes and teacher nodes with the same lecture place as the one for period \(t-1\); and let \(W'(e)\leftarrow (1 + w_1)(1 + w_2)W(e)\) for edges incident to group nodes and teacher nodes with the same lecture place as the one for period \(t-1\).

Step 4. Order the edges in a non-increasing order of \(W'(e)\) breaking ties arbitrarily. Re-index the edges as \(e_1, e_2, ..., e_{|E|}\), where \(E\) is the edge set. Stop.

3.2 Finding matchings

After obtaining an order of the edges, we find feasible matchings using the order. For a matching to be feasible in the professor-lecturer model considered in this paper, the group constraints presented earlier should be satisfied in addition to those for a typical matching for the class-teacher model. Moreover, the two additional constraints related to the lengths and start times of the lectures should also be satisfied in the MTTP. That is, edges corresponding to two-period lectures must be colored with colors associated with double periods, and an edge associated with the first period of a two-period lecture must be colored with a color associated with the first or third period of a day. To find a matching that satisfies these constraints, we keep lists of nodes whose incident edges violate one of these constraints if the edges are included in the matching in period \(t\).

In the following, we present a procedure for finding a matching in period \(t\) for edges in set \(E\). Note that the edges in \(E\) are already ordered with an edge ordering rule. First, we give additional notation used in the description.

\[
\begin{align*}
T^2 & = \text{set of teacher nodes corresponding two-period lectures} \\
E^2 & = \text{set of edges incident to } T^2 \\
e_s & = \text{edge in the } s\text{-th position in an ordered set of edges} \\
E^M(t) & = \text{set of edges in which the matching at time } t \text{ is included} \\
N'(e) & = \text{set of nodes related to edge } e, \text{i.e., } N'(e)=\{C_e, g(C_e), L_e\} \text{ if } e \text{ is incident to a class node, and } N'(e)=\{G_e, F(G_e), P_e\} \text{ if } e \text{ is incident to a group node} \\
L^N(t) & = \text{node list in period } t
\end{align*}
\]

Now, the procedure is given.

Procedure 2 (Finding matching in period \(t\) for edge set \(E\))

Step 1. Let \(s \leftarrow 1\).

Step 2. If \(e_s\) is incident to a node in \(T^2\), go to Step 5. Otherwise go to Step 3.
Step 3. If \(L^N(t)\) contains at least one of nodes in \(N'(\varepsilon_s)\), go to Step 8. Otherwise go to Step 4.

Step 4. Include edge \(\varepsilon_s\) into \(E^M(t)\), and nodes in \(N'(\varepsilon_s)\) into \(L^N(t)\). Go to Step 8.

Step 5. If \(t\) is an even number, let \(s \leftarrow s + 1\) and go to Step 2. Otherwise, go to Step 6.

Step 6. Select arbitrarily an edge from the set of parallel edges with the same pair of incident nodes as that for \(\varepsilon_s\), and let \(\varepsilon'\) denote the selected edge. If either \(L^N(t)\) contains at least one of nodes in \(N'(\varepsilon_s)\) or \(L^N(t + 1)\) contains at least one of nodes in \(N'(\varepsilon')\), go to Step 8. Otherwise go to Step 7.

Step 7. Include edge \(\varepsilon_s\) into \(E^M(t)\) and \(\varepsilon'\) into \(E^M(t + 1)\). Include nodes \(N'(\varepsilon_s)\) into \(L^N(t)\), and nodes \(N'(\varepsilon')\) into \(L^N(t + 1)\). Go to Step 8.

Step 8. If \(s = |E|\), stop. Otherwise, let \(s \leftarrow s + 1\) and go back to Step 2.

3.3 The heuristic algorithm

Now, we present a heuristic algorithm for the MTTP. In this algorithm, for each period, edges are ordered with Procedure 1 and then Procedure 2 is used to find a solution for the MTTP. This sequential execution of the procedures is done for all periods until all classes are covered. The procedure for the heuristic algorithm can be summarized as follows.

Procedure 3 (Heuristic algorithm)

Step 1. Let \(t \leftarrow 1\) and \(E^M(t) \leftarrow \emptyset\). Let \(E\) be the set of all edges.

Step 2. Execute Procedure 1 for edge ordering. Execute Procedure 2 using the edge order.

Step 3. Add colored edge sets to \(E^M(t)\) and \(E^M(t + 1)\) according to their colors, and let \(E \leftarrow E \setminus E^M(t)\).

Step 4. If \(E = \emptyset\), stop. Otherwise, let \(t \leftarrow t + 1\) and go back to Step 2.

4. Computational experiments

To evaluate the performance of the suggested algorithm, computational experiments were carried out on 300 randomly-generated problem instances. These problems were generated in such a way that they closely reflect real situations in the Korea Army Training Center. The tests include 4 variations of the heuristic algorithm that are defined by the four edge ordering rules given earlier. For the tests, data were generated randomly as follows. Here, \(U(x, y)\) and \(DU(x, y)\) denote the uniform distribution with range \((x, y)\) and the discrete uniform distribution with range \((x, y)\), respectively.

(a) The number of groups was set to 4, and the number of classes in a group was set to 4.
(b) The number of professors was selected from 3 and 4 with an equal probability, and the number of lecturers was generated from \(DU(13, 16)\).
(c) The number of teachers (professors and lecturers) that can be assigned to two-period lectures only was set to an integer closes to the number obtained by multiplying the number of all teachers by a number generated from \(U(0.2, 0.4)\).
(d) The number of teachers that can be assigned to indoor lectures only was set to an integer closes to the number obtained by multiplying the number of all teachers by a number generated from \(U(0.2, 0.4)\).
(e) The number of lectures that should be assigned to each of the lecturers who deal with two-period lectures was selected from 8 and 10 with an equal probability, while the
number of lectures to be assigned to other lecturers was generated from $DU(7,10)$. The number of lectures which each class has to get from the lecturers can be computed with this information.

(f) The number of lectures that should be assigned to each of the professors who deal with two-period lectures was selected from 16, 18 and 20 with an equal probability, while the number of lectures to be assigned to other professors was generated from $DU(15, 20)$. The number of lectures which each group has to get from the professors can be computed with this information.

Since optimal solutions of the MTTPs cannot be obtained within a reasonable amount of CPU time, solutions of the suggested algorithms were compared with each other. Note that when CPLEX 8.1, a commercial solver for integer programs, was used for the problem, it took about 400 seconds on a personal computer with Core 2 Duo Processors operating at 2.0 GHz clock speed to obtain an optimal solution for a problem with four classes and two groups. However, we could not obtain an optimal solution of the MTTP with 16 classes and 4 groups after 6 hours.

Results of the test on 300 problems are given in Table 1, which shows the averages, standard deviations as well as ranges of the makespan values and the number of setups. It also gives the number of problems for which each algorithm gave the best solutions (in terms of the makespan and the number of setups). As can be seen from the table, MHCDF-min worked best in both measures. Also, the algorithms using the MHCDF rules were better than the one using the HCFD rule. This may be because loads of edges related to classes and groups were more exactly considered in the MHCDF rules than in the HCFD rule. Among the MHCDF rules, MHCDF-min worked better than the others, especially for the measure of the number of setups, possibly because setups are reflected more in the MHCDF-min rule than in the other MHCDF rules. A series of paired t test was done on the results and it is found that there is statistically significant difference in the performance of the edge ordering rules in terms of both makespan and the number of setups.

Table 1. Results of the tests.

<table>
<thead>
<tr>
<th>Ordering rule</th>
<th>makespan (periods)</th>
<th>number of setups</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>average</td>
<td>std dev</td>
</tr>
<tr>
<td>MHCDF-max</td>
<td>192.24</td>
<td>13.99</td>
</tr>
</tbody>
</table>

To see the absolute performance of the algorithms, we compute performance ratios of the makespans. For each problem, the performance ratio (PR) of an algorithm (a variation of the algorithm defined by the ordering rules) is computed as the ratio of the makespan obtained from the algorithm to a lower bound on the makespan. The lower bound used here is given as $\max \{T^C, \max_j T_j\}$, where $T^C$ is the total number of lectures and $T_j$ is the load of (the number of lectures handled by) teacher $j$.

Table 2 shows the average and standard deviation as well as the range of performance ratios. It also shows the average CPU time required for a problem. The solutions obtained from (the variations of) the heuristic algorithm were not much far from the simple lower bounds used for the evaluation, which does not seem to be a very tight one. Solutions (makespan) of MHCDF-min were only 2.5% larger than the lower bounds on average. As in the results of Table 1, results of paired-t tests on the performance ratios also showed that
MHCDF-min was better than all the other rules at the significance level of 0.01. Also, all of the four variations of the algorithm gave the solutions in a very short computation time.

<table>
<thead>
<tr>
<th>Ordering rule</th>
<th>Average PR</th>
<th>Std Dev of PR</th>
<th>Range of PRs</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCDF</td>
<td>1.034</td>
<td>0.015</td>
<td>[1.010, 1.104]</td>
<td>0.150</td>
</tr>
<tr>
<td>MHCDF-max</td>
<td>1.029</td>
<td>0.010</td>
<td>[1.010, 1.061]</td>
<td>0.157</td>
</tr>
<tr>
<td>MHCDF-avg</td>
<td>1.027</td>
<td>0.010</td>
<td>[1.000, 1.060]</td>
<td>0.220</td>
</tr>
<tr>
<td>MHCDF-min</td>
<td>1.025</td>
<td>0.010</td>
<td>[1.000, 1.062]</td>
<td>0.155</td>
</tr>
</tbody>
</table>

5. Concluding remarks

In this paper, we considered a military training timetabling problem (MTTP) in a real military training center. In the MTTP, we took account of special characteristics of the real problem, such as constraints related to the lecture continuity of two-period lectures and setups required for changes of lecture places. Following a solution approach to the well-known professor-lecturer model for the timetabling problem, we presented a heuristic algorithm based on an edge coloring algorithm for a bipartite graph. Four variations of the algorithm were developed by varying the edge ordering rule used in edge coloring algorithm. Results of tests on a number of problem instances that reflect the real system relatively well showed that the algorithm gave relatively good solutions in a short time.

To our best knowledge, this is a first attempt to solve a real timetabling problem in a military training center with various practical considerations. Although results of this research can be actually implemented in the real system without much difficulty, this research can be extended in several ways for easier implementation and wider applicability (to timetabling problems of other educational systems). For example, further studies are needed for cases in which there are (partial) precedence relationships among various subjects and there are more than two different lecture durations for the lectures. Also, we may have to consider cases in which there are constraints related to the availability of teachers and/or material required for lectures, and in which there are more than two levels for the classes and teachers.

References