Timetabling Construction Problem (TCP)

Palgunadi
Informatika
Fakultas MIPA UNS
Jl.Ir Sutami No.36 A Kentingan
Surakarta
palgunadi@uns.ac.id

ABSTRACT

It will be described the TCP in general and proposed polinomial algorithms to solve TCP in general and different classsizes in TCP. A timetabling is a tabling of conducting lessons by teachers arranged according to the room-time when they take place that must meet a number of requirements and the timing of teaching processes must quarantee that no student and no teacher has more than one lesson at the sa me time. Let k, m, n be the number of lessons, teachers and rooms respectively, the general TCP, as well as 2-classsizes TCP, can be solved in O(k m n)

Keywords

Operation research, Timetabling construction problem, polinomial algorithm

1. INTRODUCTION

A timetabling is a tabling of conducting lessons by teachers arranged according to the room-time when they take place and they must meet a number of requirements and the timing of teaching processes must quarantee that no student and no teacher has more than one lesson at the same time.

Timetabling problem, including TCP problem, is a very hard problem. Willemen, Robertus J. (2002) said that there is no straightforward solution to solve the problem since some timetablings have many dimensions that have their own characteristics to create complexity of the problems. For example, in university timetabling course problem (UTCP) some students can be arranged in the different classes (parallel classes), different classsizes, the institution sets different curriculas, problem araised due to limitation on classroom capasities, etc.

Some researches related to TCP have been done so far. Schaerf (1995) gave an overview of the literature on TCP problems. Carter & Laporte (1996) studied on examination timetabling, and Bardadym, (1996) and Carter & Laporte, (1998) examined on university course timetabling (UTCP). De Gans (1981) described a heuristic to complete a timetable after the clustering problem has been solved. Bonvanie & Schreuder (1982) modelled timetable construction as a 0-1 programming problem and concluded that no timetable could be constructed with integer programming software at that time. Nuijten (1994), as stated in Willemen, Robertus J. (2002), introduced the time and resources constraint scheduling problem. Ferland (1998) discussed timetabling problems in the context of generalized assignment problems. Willemen, Robertus J. (2002) showed that TCP is NP Hard and NP complete problem.

In this paper, it will be described the scheme adopted from Willeman RJ (2002) and Palgunadi S (2012), and proposed the generalization of Palgunadi's algorithm (Palgunadi, S., 2012)

2. TCP

A Timetabling Construction Problem (TCP) has several elemens: teachers, lessons, classes, rooms and timeslots Given some input sets of teachers (T), lessons (L) , classes (C), rooms (R) and timeslot allocations (S). In TCP, as well as in UTCP, the planners usually conduct the timetabling with the following rules:

1. Due to the student body and classsize limitation, the student can be grouped into some parallel classes
2. Due to room capacities limitation, the planner can assign different classsizes, for example two different classsizes are the regular class (lecturing class) and lab. activities class.
3. There are room tipes, i.e rooms for lecturing (called r-room), and for lab activities (called b-room)
4. Every lesson in each class must be conducted by at least a teacher. Due to teacher skill or knowledge competencies limitation, the planner has already assigned the teachers to conduct a particular lesson in a specific class.
5. Each student must take some lessons in a specific curricula.
6. Every curricula contains some number of lessons that must be taken the students. To sake of simplicity, the planner sets only one curricula, and the teaching learning process is set to be the package system (every student in a level must take all compulsary lessons available in the corresponding level.

The planner must apply the above rules prior to the timetabling construction

The concept of object model TCP elements can then be developed based on the following scheme

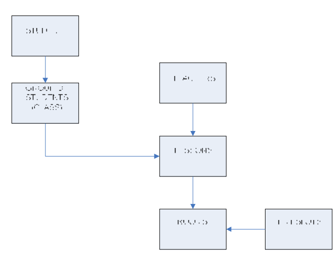

Figure 1. Relation concepts of TCP elements

Notion of the main algorithm is to transform the TCP with 5 sets of elements into the assignment problem between 2 set only and apply the exhausted search to find the maching between elements in the sets that will not arise the conflicts. To do the transformation, we observe that there will be some assignment relations needed.

a. The relation that for every lesson in a particular class must have at least a teacher (α relation), i.e α: L x C → T.

b. The relation that for any lesson in a class must be conducted at a room available in a particular timeslot (β relation), i.e β: L x C → R x S

c. The relation to check whether a room at a timeslot has alredy been occupied (γ relation) i.e γ: R x S → {0, 1}

d. The relation to check whether a teacher at a timeslot has alredy been assigned to conduct a lesson (δ relation), i.e δ : T x S → {0, 1}

Being facilitated which the four basic relations, the TCP can then be stated as 5-tuple TCP(T, L, C, R, S) that can be transformed into TCP(A,B). The objective of TCP is then to search an assigment ф: (T x L x C) x (R x S) → (0,1), says the feasiblity that a lesson l Є L for students in class c Є C taught by a teacher t ϵ T can be conducted at least (exactly) in a room r ϵ R at timeslot s ϵ S. Of course, the assigment ф must fullfill that there is no conflict araised due to limited resoures.

In creating the assigment ф, there are some conflicts may be arised due to the following conditions

a. Teacher-Lesson-Timeslot Conflict, is arised when the assigment faces rule "only a single lesson is assigned to a teacher at a particular timeslot". This condition may arise when the teacher has already been assigned to conduct another lesson (teacher-room-timeslot conflict) or he/she cannot teach at the the time since he/she is not available to teach due to having another task, for example he/she has to join faculty member staff meeting

b. Teacher-Room-Timeslot Conflict, is arised when the assigment faces rule "a teacher can only teach at a single room at a specific timeslot"

c. Class-Timeslot Conflit, is arised when the assigment faces rule "a class can only take a single lesson at a specific timeslot"

To develop the algorithm, we may observe the following facts.

a. Teacher. Let T be a set of teachers available in the TCP and |T| = m.

b. Class. Let C be a set of classes. Due to resources limitation, there are p levels with p ≤ 4 in practice, and each level there will be set the q paralel classes with q ≤ 3. We get that | C | ≤ 24

c. Lesson. Let L be a set of lessons and |L| = k. Each lesson has number of hours to be conducted weekly (called as credit hours), stated as x/y, says that the lesson should weekly be conducted x hours in lecturing (in r-room) and y hours for lab activities (in b- room). Some planners however can set that lesson with one credit hour in lab must be conducted in two hours in b-room. If so, then the lesson with x/y credit hours means that it should be conducted x hours in r-room and 2y hours in b-room (to sak of generality, we will accept this condition). In practice, the number x and y will, however, be set to 0, 1 or 2. Of course, it is impossible to set both numbers x and y to be 0, i.e x=0 and y=0.

Since the timeslot is allocated hoursly, we then can reduce the conflicts that might be arised with the following step, Replace every lesson l ϵ L with x/y credit hours with l r1 ,l r2, ,..., l rx ϵ Lr and l b1 ,l b2,..., l b2y ϵ Lb, Observe that, in practice, (i) x, y ≤ 2; (ii) the set L will become L= Lr U Lb, where Lr || Lb, (iii) every li Є L be already set hoursly, and (iv) |L| = |Lr| + |Lb| ≤ 6 k

d. Teacher-Class-Lesson. We can reduce the conflicts using the following step. Let A be equal to T x C x L, and there exist a-assignment a(t,c,l) says that a lesson l taught by teacher t for class c. The a-assignment is important to guarantee that a lesson l for class c can only be taught by a teacher t and must be determined before the planner conducts the TCP.

The number of operations to calculate the a-assigment is then at most 6*k*24*m = 144 k m operations

e. Room. Let R be set of rooms, then R can then be divided into two disjoint sets, i.e R = Rr U Rb where Rr || Rb . Let |Rr| = nr and |Rb| = nb. The cardinality of R will be n = nr + nb. Since each lesson may need a specific tool that may not belong to any room rb Є Rb, then Rb should be set into some disjoint sets. The later condition, however, will be ignored in this model.

f. Timeslot. Let S be a set weekly timeslot availabe in each room, where maximum number of slot is set to be 11 and the rooms are available for 5-days a week. So |S| ≤ 55. For any s ϵ S, s is then coded into 3-digit code according to corresponding day (first digit) and slots (the last two digits). For example code s = 307 means timeslot for the third day at timeslot 7.

g. Room-Timeslot. To the sake of simplicity, let B be R x S the set of b-assignment b(r,s) says the room r at slot s. This b-assignment is important to simplify to handle the room-timeslot resources. It is clear that b-assignment can be obtained in at most 55 n number of operations

Having the above model, the objective of TCP becomes to search an assigment ф: A → B where A=T x C x { Lr U Lb } and B = {Rr U Rb } x S. It is clear that If there exists an assigment ф: A → B in TCP, then assigment ф is a one-one relation but not onto, and |B| ≥ |A|

To solve the TCP, we need several matrices and sets as follows.

a. A boolean matrix TS: T x S → {0,1} says that whether a teacher t has already been assigned to teach at timeslot s. The matrix can be created initial and set its elements to 0. If TS(t1, s) = TS(t2, s) = 1 then t1 = t2 unless the Teacher-Timeslot conflict will occur. Since some teachers may have been assigned to teach in another department or the teachers will join in a regular meeting, then some teacher timeslots should be blocked, ie TS(t,s)'s are set to 1.

b. A boolean matrix CS: C x S → {0,1} says that whether a class c has already been assigned to take a lesson at timeslot s. The matrix can be created initial and set its elements to 0. It is clear that if (c1, s) = (c2, s) = 1 then c1 = c2 unless the Class-Timeslot conflict will occur.

c. Four set of the assigments. Let AS be set of elements in T x L x C that have been schedulled. Let ANS be set of elements in T x L x C that have not been schedulled yet. Let BS be set of elements in R x S that have been occupied, and let ABS be set of elements in ф th at have been assigned correctly

The main algorithm to solve the TCP is described as follows

Algorithm generalTCP (T, C, L, R,S)

(1) Create the a-assignment (A), where A = T x C x { Lr U Lb } and split A into two disjoint set Ar = T x C x Lr and Ab = T x C x Lb

(2) Create the b-assignment (B), where B = {Rr U Rb } x S and split B into two disjoint set Br = Rr x S and Bb = Rb x S

(3) intialize the matrices TS and CS, AS = {} , ANS = {} BS = {}, ABS = {}

(4) while (there is an unexmined element in A) do begin

(4.1) select a be unexamined element Є A; scheduled = false;

(4.2) if a Є Ar then begin
{conducted in lecturing room}

(4.2.1) while (not(scheduled) or (there is an unexmined element in Br) do begin

(4.2.1.1) select b be an unexamined eleme nt Є Br

(4.2.1.2) if TS(a.t, b.s) = 0 and CS(a.c, b.s) = 0 then begin

{ check the conflict}

scheduled = true; { no conflict}

AS = AS + {a} ; BS = BS + {b} ; ABS = ABS + {(a,b)};

TS(a.t, b.s) = 1 ; CS(a.c, b.s) = 1;

A = A – { a } , B = B – {b}; end

else select b.next ; {try next room-timeslot}

end

(4.2.2) If not(scedulled) then begin

ANS = ANS + {a} ;

A = A – {a} end

else {conducted in lab }

(4.2.3) while (not(scheduled) or (there is an unexmined element in Bb) do begin

(4.2.3.1) select b be an element Є Bb

(4.2.3.2) if TS(a.t, b.s) = 0 and CS(a.c, b.s) = 0 then begin

scheduled = true;

AS = AS + {a} ; BS = BS + {b} ; ABS = ABS + {(a,b)};

TS(a.t, b.s) = 1 ; CS(a.c, b.s) = 1;

A = A – { a } , B = B – {b}; end

else select b.next ;

end;

(4.2.4)) If not(scedulled) then begin

ANS = ANS + {a} ;

A = A – {a} end

end;

Observe that,

- if the algorithm sucessfully solves the TCP, by the end of the algorithm, the set ANS is empty, says that all lessons have already been sc heduled.
- The algorithm will produce four sets, teacher scheduled (TS), class scheduled (CS) and the TCP assignment (ABS), the unscheduled lesson sets (ANS)

The complexity of the algorithm can be calculated as follows

- Step (1) can be done at most | A | operations = 144 km operations
- Step (2) can be done at most | B | operations = 55 (|Rr | + |Rb |) . If (|Rr | + |Rb |) = n then step (2) can be done at most 55 n operations
- Step (3) can be done at most 55 (|T| + |C|) + 3 operations = 55 (k + m) + 3 operations
- Step (4.1) is only 1 operation,
- Step (4.2) can be done at most at 55 * n *11 operations = 605 n operations.
- Step (4) can be done at most at = 144 km * 605 n operations. Step (4) can be done at most in 87120 kmn operations.
- Total number of operations is then 144 km + 55 n + 55 (k + m) + 3 +87120 kmn. In “big Oh notation” we get O(kmn). Consequently, the following theorem holds

Theorem 1. Let k, m, n be the number of lessons, teachers and rooms respectively. The TCP can be solved in O(k m n),

The 2-classsize TCP

In the above TCP, the planner has not considered the conditions that the classsize imay be adjusted due to different capasities of r-room and b-room. If the r-room capasity is set to be double than that of b-room, then it becomes clear that the main algorithm should be minor modified to deal with latter condition. It is clear that this modification can be done in constant operations. So the complexity of the algorithm can remain to be polinomial.

Observe that each class c Є Cr becomes two classes c1 dan c2 Є Cb. So, The following conditions will occur

- if CS(c,s) = 1, then CS(c1,s) = 1 and CS(c2,s) = 1
- if CS(c1,s) = 1 or CS(c2,s) = 1, then CS(c,s) = 1

By adding the above conditions into the step (4.2.1.2) and step (4.3.1.2) in the main algorithm we still can use the main algorithm to solve the 2-classsize TCP. It is then obvious that the following algoritm holds

Theorem 2. Let k, m, n be the number of lessons, eachers and rooms respectively. The 2-classsize TCP can still be solved in O(k m n)

3. CONCLUSIONS

It can be concluded that the TCP can be solved in polinomial time. In fact, however, the algorithms have not been implemented yet. A high constant at the number of operations on complexity of the algorithm may cost the run times of the algorithms become very slowly. Some possible future researches are: examining the run time of the two TCP algorithms, developing the new algorithms to deal with adding the constraints due to(i) limited tools to conduct a parti cular lab activity lesson at a specific lab room, (ii) 3-classsize TCP

4. BIBLIOGRAPHY

[1] BARDADYM, V.A. [1996], Computer-aided school and university timetabling: The new wave?, in: E.K. Burke and P. Ross (eds.), Practice and Theory of Automated Timetabling, First International Conference, Selected papers, Lecture Notes in Computer Science 1153, Springer, 22–45.

[2] BONVANIE, M.C., AND J.A.M. SCHREUDER [1982], Model lesroosters vakkenpakket VWO, Technical Report 411, Universiteit Twente, Enschede, The Netherland s.

[3] CARTER, M.W., AND G. LAPORTE [1996], Recent developments in practical examination timetabling, in: E.K. Burke and P. Ross (eds.), Practice and Theory of Automated Timetabling, First International Conference, Selected papers, Lecture Notes in Computer Sc ience 1153, Springer, 3–21.

[4] DI GASPERO, L., AND A. SCHAERF [2001], Tabu search techniques for examination timetabling, in: E.K. Burke and W. Erben (eds.), Practice and Theory of Automated Timetabling III, Third International Conference, PATAT 2000, Selecte d papers, Lecture Notes in Computer Science 2079, Springer, 104–117.

[5] FERLAND, J.A. [1998], Generalised assignment-type problems: A powerful modeling scheme, in: E.K. Burke and M.W. Carter (eds.), Practice and Theory of Automated Timetabling II, Second International Conference, PATAT '97, Selected papers, Lecture Notes in Computer Science 1408, Springer, 53–77

[6] GANS, O.B. DE [1981], A computer timetabling system for secondary schools in the Netherlands, European Journal of Operational Research 7, 175–182.

[7] KINGSTON, J.H. [2001], Modelling timetabling problems with STTL, in: E.K. Burke and W. Erben (eds.), Practice and Theory of Automated Timetabling III, Third International Conference, PATAT 2000, Selected papers, Lecture Notes in Computer Science 2079, Springe r, 309–321.

[8] LEWANDOWSKI, G. [1994], Practical implementations and applications of graph coloring, Ph.D. thesis, University of Wisconsin-Madison, U.S.A.

[9] NUIJTEN, W.P.M. [1994], Time and resource constrained scheduling: A constraint satisfaction approach, Ph .D. thesis, Technische Universiteit Eindhoven, The Netherlands.

[10] PALGUNADI, S., [2012]. A New Algorithm on Timetable Construction Problem (TCP). A Case study at Vocational Education on Informatics Engineering (D3 TI) FMIPA University Sebelas Maret. The International Conference on Entrepreneur University: Wikinomics Strategy to Raise the Local Wisdom. FMIPA UNS. 2012. Selected papers.

[11] REIS, L.P., AND E. OLIVEIRA [2001], A language for specifying complete timetabling problems, in: E.K. Burke and W. Erben (eds.), Practice and Theory of Automated Timetabling III, Third International Conference, PATAT 2000, Selected papers, Lecture Notes in Computer Science 2079, Springer, 322–341.

[12] SCHAERF, A. [1995], A survey of automated timetabling, Technical Report CSR9567, CWI, Amsterdam, The Netherlands.