2012 International Conference on Industrial and Intelligent Information (ICIII 2012)
IPCSIT vol.31 (2012) © (2012) IACSIT Press, Singapore

A Novel Combinational Algorithm for Solving the Examination Timetabling Problem

Roohollah Etemadi [1, +], Nasrollah Moghadam charkari [2]

[1]Department of Electrical and computer engineering, Islamic Azad University Bonab Branch, Bonab, Iran

[2] Department of Electrical and computer engineering, Tarbiat Modarres University, Tehran, Iran

Abstract. The examination timetabling problem is the allocation of a certain number of examinations to a limited number of timeslots in compliance with a set of constraints. In this paper, a new two-phase algorithm has been presented based on clustering and tabu search algorithm to solve the timetabling problem of examinations. The recommended algorithm has two phases. In the first one, a set of examinations are clustered into appropriate clusters using the clustering algorithm, and in the next one, an appropriate timeslot is presented for each obtained cluster using the tabu search algorithm. The results obtained through the evaluation of the proposed algorithm on the Toronto standard data sets and its comparison with the selected famous algorithms from 1996 to 2010 in this field indicate the efficiency and high speed of the proposed algorithm compared with the other ones, so that the proposed algorithm gains appropriate and acceptable timetabling for the examinations, much faster than the other ones.

Keywords: Clustering Algorithm, Tabu Search, Examination Timetabling Problem.

1. Introduction

The examination timetabling problem is the allocation of a specific set of exams to a limited number of timeslots with the capacity of rooms for each timeslot, with regard to a set of necessary constraints. The set of constraints in the examination timetabling problem are divided into two groups, which are: Hard Constraints and Soft Constraints. Hard constraints are those that should definitely be satisfied in the examination timetabling. For example, two examinations which are incompatible with each other (they have students in common) should not be timetabled within a similar timeslot. The timetabling which satisfies all hard constraints is called possible timetabling. The constraints, which are taken into consideration in order to increase timetabling favorability, are called soft limitation. For example, examination timeslot, which are incompatible with each other, should be at least more than 5 timeslots. The quality of a possible timeslot is usually measured based on the violation or observation of the soft constraints. For more information about these constraints refer to reference [1], [3].

During recent years, different methods have been presented for solving the examination timetabling problem, among which we can mention graph-based methods [2], [4]. Other methods used to solve the examination timetabling problem were local search-based ones (tabu search and cooling metals and searching the most adjacent neighbor) [7], [12], [14] , population-based methods (evolutionary algorithms, the Memetic Algorithm,

+ Corresponding author. Tel.: +98 914 102 5279.
E-mail address: r.etemadi@bonabiau.ac.ir and roohalahetemadi@gmail.com

the ant colony algorithm, the artificial immune system algorithm) [5-6],[9], [11], [13], the Meta Heuristics [8] and decomposition and clustering techniques [4].

The rest of this paper has been prepared as follows: in part 2 the proposed algorithm is discussed; part 3 is about the evaluation of the proposed algorithm and includes the results of the proposed algorithm implementation on the introduced test data and their comparison with the results of algorithms in this field; and, part 4 includes conclusion of the paper.

2. Proposed Algorithm

ESTSA algorithm has two main phases for examination clustering. In the first one, through using the proposed clustering algorithm (ESA), the examination set is clustered into appropriate clusters, which are less than or equal to the number of timeslots. In the second phase, through using the proposed tabu search algorithm (TSA), an appropriate timeslot is allocated to each examination cluster. Algorithm (1) shows main phases of the proposed ECTSA algorithm.

2.1 Proposed Examination Clustering Algorithm (ECA)

The strategy of the ECA proposed algorithm for the examination clustering is that the examination set, with regard to the hard constraints, is changed into an equivalent graph and vertices of equivalent graph are clustered in a way that no two arbitrary vertices are in a contiguous (adjacent) cluster (hard constraint satisfaction). After clustering examination graph vertices, for timetabling, we can use examination clusters instead of exams. In other words, instead of examination timetabling, we timetable examination cluster in the determined timeslots. As the number of clusters is far less than the number of examination, the search space will be dramatically reduced when using the clusters for timetabling. The reduction of problem search space can affect the proposed tabu search algorithm for timetabling examinations. Before presenting the proposed algorithm of ECA, following definitions are provided.

Definition 1: Consider the set of $X = \{x_1, x_2, ..., x_n\}$ which includes n objects. The purpose of clustering is categorization of n objects into k clusters so that each cluster has the following conditions:

1) $C_1 \cup C_2 \cup ... \cup C_k = X$ 2) $C_i \neq \phi \quad i = 1...k$ 3) $C_i \bigcap C_j = \phi \quad i \neq j, 1 \leq i, j \leq k$

The problem of finding the best mode for clustering n objects to K clusters is considered as an NP-Complete and complicated problem.

Definition 2: We suppose the weighted and undirected graph of G(V,E) as the equivalent graph with examination set in which $V = \{v_1, v_2, ..., v_n\}$ set is called graph vertices and $E = \{e_1, e_2, .., e_m\}$ set is called G graph edges. Each vertex of the graph corresponds to one examination. The edges between the two optional vertices indicate a student in common between the two corresponding exams. The vertices of the two ends of that edge and the weight of edge can show the number of students in common. The aim of examination set clustering is grouping the vertices in V to k clusters such as $C = \{C_1, C_2, ..., C_k\}$ in a way that the following relations are dominant:

1) $C_1 \cup C_2 \cup ... \cup C_k = V$ 2) $C_i \neq \phi \quad i = 1,2,...,k$

3) $C_i \bigcap C_j = \phi \quad i \neq j, 1 \leq i, j \leq k$ 4) $\forall u, v \in C_i : (u,v) \notin E, 1 \leq i \leq k$

In the above definition, the number of clusters (k) equals the number of timeslots and condition 4 ensures that no two optional exams own a student in common in one optional cluster (hard constraint satisfaction).

Definition 3: Non-conflict set for a Cluster (NC-Set): This set is defined for a cluster and consists of all non-clustered vertices' set of the examination graph which do not have direct edges with the vertices in the pertinent cluster. In other words, it consists of a set of non-clustered vertices of examination graph that can be added to the pertinent cluster.

Definition 4: Similarity Criteria between Nodes and clusters: This criterion shows similarity of one node of examination graph with a cluster and is defined as the relation (1).

$$sim(node, C) = \begin{cases} 1 - \frac{conflit\_Number(node, NC\_Set(C))}{|NC\_Set(C)|} & if\ node \in NC\_Set(C) \\ 0 & if\ node \notin NC\_Set(C) \end{cases} \quad (1)$$

In the above relation, $sim(node, C)$ shows the similarity of node with the cluster C. If the node does not exist in non-conflict cluster of C ($node \notin NC\_Set(C)$), its similarity with the cluster C is considered as zero. Otherwise, the number of common edges between the node and nodes in $NC\_Set(C)$ is calculated, according to which the similarity between the node and the cluster C is determined. It is clear that the lesser the number of these edges, the more the similarity of the node with the cluster C will be. The node can be added to the cluster in one phase if its similarity with the pertinent cluster is more than that of the other nodes. The value of $sim(node, C)$ will be between 0 and 1 and the node with the similarity of 0 and with a certain cluster cannot be added to that cluster.

Definition 5: A set of seed nodes: It is a set of examination graph nodes which form a complete sub-graph. In other words, as these nodes have an edge in common, they cannot be added to the same clusters. Therefore, these nodes are recognized at the beginning of the work and they are considered as the initial centers of the clusters.

```
Algorithm1. pseudo-code of the Proposed ECTSA Algorithm
1. Input: Weighted Graph G, which Equal with Enrolled Examination and the number of timeslot.
2. Output: scheduling of exams
3. Begin
4. SetOfCluster    ECA ( G );     /* section 2.1 */
5. TSA(SetOfCluster,timeslot);     /* section 2.2  */
6. End
```

Algorithm2. pseudo-code of the Proposed Exam Clustering Algorithm (ECA)

1. **Input**: Exam Graph G(V,E) and number of timeslot (k)
2. **Output**: clustering Graph G with H clusters (H $\geq$ k)
3. **Begin**
4. Find seed nodes $N = \{n_1, n_2, ..., n_h\}$
5. Create $|N|$ Clusters $C = \{C_1, C_2, ..., C_{|N|}\}$ and add $n_i \rightarrow C_i$.
6. Set reminder nodes RN=V-N
7. $num\_c \leftarrow |N|$
8. While $|RN|$ $!= 0$ Do
    - 8.1. calculate similarity for each node in reminder set RN with each cluster in C
    - 8.2. find max $sim(m_i, C_l), m_i \in Rn, C_l \in C$
    - 8.3. If $Sim(m_i, C_l) > 0$ Then add m_i to C_l.
    - 8.4. If $Sim(m_i, C_l) = 0$ Then Create new Cluster such as C_{num_c+1} and $m_i \rightarrow C_{num\_c+1}$
    - 8.5. num_c++
9. $H \leftarrow num\_c$
10. **End**

Algorithm3. pseudo-code of the Proposed Tabu search Algorithm (TSA)
1. **Input**: Exam clusters and number of timeslot (k), Repeat (r).
2. **Output**: Exam timetabling
3. **Begin**
4. Crate possible solution(S).
5. Sbest=S; FitSbest=Fitness(Sbest);
6. Crate tabu list, $i \leftarrow 1$,
7. While Condition( ) and i< r Do
7.1. S = Move(S);
7.2. if (Fitness(S) < Fitbest)
Sbest=S; Fitbest=Fitness(Sbest);
7.3. Update tabu list.
7.4. i++
8. Return Sbest;
9. End

The main idea in ECA algorithm for the examination graph clustering is that, in each examination set graph, first a complete sub-graph should be selected and each of its vertices should be considered as the centers of the clusters. Then, by continuation of the algorithm, the remaining of the graph nodes is added to the appropriate clusters so that the nodes within a cluster do not have a cluster with direct edges (hard constraint satisfaction of the examination timetabling problem). In case, at one stage, none of the non-clustered nodes is able to be added to the existing clusters, a new cluster is created and one of the nodes is added to that. This will continue as long as all the nodes are added to the appropriate clusters. Algorithm (2) shows the execution steps of the ECA algorithm.

2.2 Proposed Tabu Search Algorithm for Timetabling the Clusters (TSA)

In the proposed ECTSA algorithm for the examination timetabling, tabu search is used for the examination clustering timetabling and timetabling improvement. The proposed TSA algorithm performs examination cluster timetabling in a way that the satisfied soft constraints reaches a maximum. Algorithm (3) shows the proposed TSA algorithm for the cluster timetabling. Each of the components of this algorithm will be studied in more details later.

The TSA algorithm satisfies the specified soft constraint for allocation of timeslots to the examination clusters. In order to satisfy the specified soft constraint, the clusters with students in common should be placed at the timeslots with maximum distance between them. Therefore, according to the closeness of the timeslots of the students' examinations in each solution, we can determine a fine and calculate a total fine based on it. With respect to what explained above, the fine can be calculated based on relation (2) [2]:

$$fitness = penalty = \sum_{k=1}^{m-1} \sum_{h=k+1}^{m} (w_i \times s_{kh}) / s \qquad i \in \{0,1,3,3,4\} \tag{2}$$

In above Relation, m is number of exams, S is number of students, S_{kh} is number of number of common students of e_k, e_h exams. w_i is the amount of the determined penalty to timetable of the two exams of e_k, e_h with the timeslot equals to i, which is calculated by Relation (3):

$$1 \le i = |t_k - t_h| \le 5 \qquad w_i = 2^{|5-i|} \tag{3}$$

In above Relation, t_k, t_h is the timeslot of e_k, e_h exams. With respect to Relation (3), the more exams with common students are timetabled within the timeslots, the more penalty will be received by them.

With respect to Relation (2), the less the amount of Relation (2) is for a solution, the higher is the competence of its solution. In other words, if the amount of Relation (2) for a solution is less, it shows that the timetabling specified by the solution will be better and it will further satisfy the specified soft constraint.

3. Experimental Results

Toronto Standard Data set has been used to test the proposed algorithm. To use this Data set, we can refer to http://www.asap.cs.nott.ac.uk/resources/ data.html. The proposed algorithm has been coded by C#.net2005 language and has been run on Pentium IV computer system, a processor with 3.06GHz speed and 512 MB RAM. The results obtained from the execution of the proposed algorithm are reported in Tables (1) and (2).

Results of the proposed ECTSA algorithm together with the reported results of 11 famous algorithms from 1996 to 2010 have been presented in Table (1). The specified amounts in Tables have been calculated based on Relations (2) and (3). In other words, the reported amounts in these Tables are the penalty of each timetabling that have been reported based on Relation (2), due to the violation of exams timetabling problem soft constraint. Regarding the explanations offered about Relation (2), the smallness of the amount of the Relation for a timetabling indicates that its timetabling is appropriate and the soft constraint of the problem (exams timetabling of a student with maximum timeslot from each other) is more satisfied.

The data set for which the selected algorithms in the specified reference have not reported any results, no value has been written. The first column of Table (1) contains timetabling algorithms and the remaining columns indicate the reported results of the corresponding algorithms for the sample of the test data under study. The last row of Table (2) shows the results obtained by the proposed algorithm on the data sample. As Table (1) shows, the proposed ECTSA algorithm has obtained an appropriate timetabling in all test data samples so that the penalty resulted from the proposed algorithm were better than the results of the majority of algorithms under study and it was close to the best results obtained by other famous algorithms in this field.

Runtime of the proposed algorithm has been presented in Table (2) for timetabling of the sample test data. Run-time of the presented algorithms has not been reported in most references. To show the run-time speed of the presented algorithm, we have compared its run-time with two (Pillay et al. 2010) IGA and (Burke et al.2010) VNS algorithms. IGA algorithm has been coded by Java language and has been run on Apple iMac computer system with Dual Core of 2.16 MHz and 1 GB RAM memory and VNS algorithm has been run on a system with Athlon 750MHz processor.

Table 1. The best experimental results obtained from running the proposed algorithm and the related famous algorithms in this field on Toronto Data Set

Approaches/ Techniques	Data Set and Penalties											
	car-f-91	**car-f-92**	**ear-f-83 I**	**hec-s-92I**	**kfu-s-93**	**lse-f-91**	**Rye-f-92**	**sta-f-83I**	**tre-s-92**	**uta-s-92I**	**ute-s-92**	**yor-f-83I**
Carter et al. 1996[4]	7.1	6.2	36.4	10.8	14.0	10.5	7.3	161.5	9.6	3.5	25.8	41.7
Caramia et al. 2001[5]	6.6	6.0	**29.3**	**9.2**	13.2	**9.6**	**6.8**	158.2	9.2	3.2	**24.4**	**36.2**
Burke et al. 2003[6]	4.65	4.1	37.05	11.54	13.9	10.82	-	168.73	8.35	3.2	25.83	37.28
Burke et al. 2004 [7]	4.8	4.2	35.4	10.8	13.7	10.4	8.9	159.1	8.3	3.4	25.7	36.7
Yang et al. 2005[8]	**4.5**	**3.93**	33.7	10.83	13.82	10.35	8.53	158.35	7.92	**3.14**	25.39	36.35
Burke et al. 2006[9]	4.6	4.0	32.8	10.0	**13.0**	10.0	-	159.9	**7.9**	3.2	24.8	37.28
Abdullah et al. 2007[10]	5.2	4.4	34.9	10.3	13.5	10.2	8.7	159.2	8.4	3.6	26.0	**36.2**
Eley 2007[11]	5.2	4.3	36.8	11.1	14.5	11.3	9.8	157.3	8.6	3.5	26.4	39.4
Caramia et al. 2008[12]	6.6	6.0	**29.3**	**9.2**	13.8	**9.6**	-	158.2	9.4	3.5	**24.4**	**36.2**
Qu et al (AGH-GHH.)2009 [2]	5.09	4.26	35.48	11.46	14.68	11.2	-	158.28	8.51	3.15	27.9	40.49
Pillay et al. 2010[13]	4.9	4.2	35.9	11.5	14.4	10.9	9.3	157.8	8.4	3.4	27.2	39.3
ECTSA (Proposed Algorithm)	6.2	5.1	38.52	11.54	15.5	13.6	12.9	157.40	8.79	4.18	29.7	40.1

As Table (2) shows, the proposed algorithm has not been comparable with these two algorithm in terms of speed and it was faster by far as the proposed algorithm has obtained acceptable results within much less period of time.

For example, the proposed ECTSA algorithm timetables sta-f-83 data sample with the penalty of 157.40 within 0.075 seconds, whereas IGA algorithm timetables the same data sample with the penalty of 157.8 within 469 seconds.

Table 2. The best results obtained and run-time of the proposed algorithm on Toronto Test Data and comparing it with well-known algorithms in this field

Data set	Penalty Results					
	ECTSA		IGA(2010) Pillay et al. 2010[13]		VNS(2010) Burke et al.2010[14]	
	Best	CPU Time(s)	Best	CPU Time(s)	Best	CPU Time(s)
ear-f-83I	38.520	0.589	35.9	751	**32.8**	3084
hec-s-92I	11.544	0.220	11.5	448	**10.0**	28
kfu-s-93	15.529	0.397	14.4	3168	**13.0**	673
lse-f-91	13.631	0.260	10.9	2863	**10.0**	345
sta-f-83I	**157.40**	0.079	157.8	469	-	-
tre-s-92	8.798	1.238	8.4	1121	**7.9**	218
ute-s-92	30.138	0.188	27.2	663	**24.8**	73
yor-f-83I	40.124	1.216	39.3	552	**34.9**	126
rye-s-93	12.99	0.563	9.3	4320	**-**	-
car-f-92	5.143	1.372	4.2	4260	**3.9**	1686
car-f-91	6.269	1.796	4.9	5880	**4.6**	3084
uta-s-92I	4.182	2.557	3.4	3639	**3.2**	2040

4. Conclusion

In this paper, a two-phase algorithm based on clustering and tabu search algorithm (ECTSA) has been proposed to solve exams timetabling problem. By studying the available algorithms to solve exams timetabling problem, we will find out that it is the first time the idea of exams clustering for timetabling them has been discussed in this paper. The main idea of the proposed algorithm is that before exams timetabling within timeslots, first, exams sets are clustered to appropriate clusters with fewer or equal number to the timeslots. In the second phase, using tabu search algorithm, an appropriate timeslot is allocated to the existing exams within a cluster. In other words, all exams within a cluster are timetabled at one timeslot. After allocation of appropriate timeslots to each of the clusters, structure of clusters is changed to increase the quality of timetabling.

Toronto Standard Data Set has been used to measure the efficiency of the proposed algorithm. The results reported from running the proposed algorithm indicate the efficiency of the proposed algorithm compared with other algorithms under study since the proposed algorithm obtained acceptable results within far less time.

5. References

[1] R. Qu · E.K. Burke · B. McCollum · L.T.G. Merlot · S.Y. Lee," A survey of search methodologies and automated system development for examination timetabling ", J Sched (2009) 12: 55–89.

[2] R. Qu, Edmund K. Burke, Barry McCollum ," Adaptive automated construction of hybrid heuristics for exam timetabling and graph coloring problems", European Journal of Operational Research 198 (2009) 392–404.

[3] Garey, M. R., & Johnson, D. S. (1977). "Computers and intractability a guide to the theory of NP-completeness", W.H. Freeman.

[4] Carter, M. W., Laporte, G., & Lee, S. Y. (1996). "Examination timetabling: algorithmic strategies and applications". Journal of Operational Research Society, 47(3), 373–383.

[5] Caramia, M., Dell'Olmo, P., & Italiano, G. F. (2001). "New algorithms for examination timetabling". In S. Naher & D. Wagner (Eds.), Lecture notes in computer science: Vol. 1982. Algorithm engineering 4th international workshop, proceedings WAE 2000 (pp. 230–241). Berlin: Springer.

[6] Burke, E. K., & Newall, J. P. (2003). "Enhancing timetable solutions with local search methods". In E. K. Burke & P. De Causmaecker (Eds.), Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference (pp. 195–206). Berlin: Springer.

[7] Burke, E. K., Bykov, Y., Newall, J. P., & Petrovic, S. (2004). "A time predefined local search approach to exam timetabling problems". IIE Transactions, 36(6), 509–528.

[8] Yang, Y., & Petrovic, S. (2005). "A novel similarity measure for heuristic selection in examination timetabling". In E. K. Burke & M. Trick (Eds.), Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V: selected papers from the 5th international conference (pp. 377–396). Berlin: Springer.

[9] Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R. (2006). "Hybrid variable neighborhood approaches to university exam timetabling (Technical Report NOTTCS-TR-2006-2)". School of Computer Science, University of Nottingham.

[10] Abdullah, S., Ahmadi, S., Burke, E. K., & Dror, M. (2007). "Investigating Ahuja–Orlins large neighborhood search for examination timetabling". OR Spectrum, 29(2), 351–372.

[11] Eley, M. (2007)." Ant algorithms for the exam timetabling problem". In E. K. Burke & H. Rudova (Eds.), Lecture notes in computer science:Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference (pp. 364–382). Berlin: Springer.

[12] M. Caramia, P. Dell'Olmo, G.F. Italiano, "Novel local-search-based approaches to university examination timetabling", INFORMS Journal of Computing 20 (1) (2008) 86–99.

[13] N. Pillay, W. Banzhaf, "An informed genetic algorithm for the examination timetabling problem", Applied Soft Computing 10 (2010) 457–467.

[14] E.K. Burke, A.J. Eckersley, B. McCollum, S. Petrovic, R. Qu, "Hybrid variable neighbourhood approaches to university exam timetabling", European Journal of Operational Research 206 (2010) 46–53.