# The Perception of Interaction on the University Examination Timetabling Problem

J. Joshua Thomas Ahamad Tajudin Khader Mohammed Al- Betar

Bahari Belaton

Artificial Intelligence Lab, School of Computer Sciences, University Sains Malaysia, Penang, Malaysia

joshopever@yahoo.com, tajudin@cs.usm.my, mohbetar@cs.usm.my

bahari@cs.usm.my

**Abstract.** In real-world perspective, educational institutions have come across a mixture of formulation for the examination timetabling problem and still semi- automated scheduling systems are in practice. In this paper, we look into the knowledge abstractions techniques to reduce the complexity of problem solving for university examination timetabling problem. The methods consists of *recapitulate, visual analysis heuristic (VAH), and specification.* The recapitulation groups the successive components, thus reducing the size of the problem. The clustering heuristics, partitioning the problem into easy and difficult components interacting through abstracted pools. The hierarchy of pools allows the user to intervene in conflict resolution at the most appropriate level of abstraction. The specification simplify the decouple components to aid the user in assessing the stiffness of the problem. We propose an algorithm that interleaves these processes. The combinations of these three techniques are evaluated with the real-world examination timetabling room allocation problem scenario. The merits of our approach are minimizing the need for backtracking, provides interactive visual models framework to understand the conflict resolution and offering a comprehensive direction to feasible solution.

**Keywords:** Keywords: Examination Timetabling, Visualization, Conflict resolution

# 1 Introduction

The common feature of constraint satisfaction problems is the fact that each variable ranges over a finite domain. Problems in this class are theoretically desirable. A simple algorithm can be exhibited that eventually finds the solutions, if any, and terminates. The real problem is efficiency that is, finding effective feasible solution. Scheduling, assignment, planning are closely related problem. The main reason is they intercede at different time scales and we are looking at that class of problem in the paper.

The scheduling of examinations to time periods is a problem faced by many educational institutions at the end of the academic semester (Lewis et al. 2005a, Burke and Kingston et al. 2004). The basic form of the examination timetabling is tackled by assigning a set of examination to predetermined time periods so as to satisfy the predetermined constraints. The constraints are either hard or soft. The former must be satisfied in order to come up with a feasible timetable while satisfying the soft constraints is desired by not essential. The conventional research objective is to minimize the total number of soft constraint violations in a feasible timetable.

Assorted and modern approaches such as hyper heuristics (Burke, McCollum et al.2007), tabu search (Chiarandini et al.2006), evolutionary algorithms (Doyle 1979, Cote 2005, Burke and Newall 1996) and simulated annealing (McCollum and McMullan 2008), particle swarm algorithms (Erben 2001), and harmony search algorithms (Al-Betar et al. 2008) are in the list on solving the examination timetabling problems. Graph based techniques are beneficial to constructing solutions by ordering the exams that have not yet been scheduled according to the obvious difficulty in scheduling that exam into a feasible timeslot. In the newest dataset establish by ITC-2007 or other real world case, finding a feasible solution by using graph colouring heuristics becomes implausible.

The work carried out by (Lewis and Paechter 2005a, Lewis and Paechter 2005b, Lewis et al. 2007) can be good evidence that the real-world examination instances normally cannot be feasibly tackled using classical graph colouring approaches. The author employed a grouping genetic algorithm in order to solve the first-ordering constraints against 60 test problem instances establish by him for post-enrolment course timetabling. Eventually, the researcher was not able to find feasible solutions for all problem instances.

In fact, the dataset established by the first international timetabling competition (TTComp2002) for course timetabling problem has been not concern in the difficulty of finding a feasible solution. It is awarded the participants based on those who obtained a feasible solution with the least number of sort constraint violations (Chiarandini Birattari et al 2006). Lately, the attentions of the timetabling research communities have been turned toward closing the gap between the fabrication datasets using in research and the real-world dataset especially when the influential research carried by (McCollum et al. 2007) was published. As such, the newer dataset released by ITC-2007 (McCollum et al. 2009) for the post enrolment was more realistic in which the term 'distance to feasibility ' is introduced as a factor of evaluating the solution obtained and thus the competitor are win when they find a feasible schedule for some problem instances.

In extending and explaining the techniques behind the Visual Analysis Heuristics (VAH), it is beneficial to look at the reasons for investigating the topic, into a broader research context. It is essential to clearly define the subject of interactive visualization as a basic formation. In real-life examination scheduling it becomes highly complex making valid solutions where the visual representations can be used as a guidance measure to reduce the complexity. Our previous work has shown a visual analysis framework on the pre-processing (J.J. Thomas et al. 2008, J.J. Thomas et al. 2009) over the examination timetabling problem.

There are a number of software commercial systems available for examination scheduling (Erben 2001) each of which use apt user interfaces which allow user to address the design and implementation issues of the search heuristics in a more standard way. These computing system need human knowledge to intervene few processes to ease the process and learn visually.

The interest in this research work is on solving the examination allocation problem, an assignment usually to one or more human machinist. For example human schedulers or human decision makers who applied a heuristic assignment procedure, based on the knowledge and with little guidance from computer software to avoid clashes in relation to solution the measurement of evaluation function is not feasible or ideal and it was allowed to adjust the weight to enable the solution (Cumming et al. 2006). In this paper we cast the examination timetabling problem and the organization of the paper is the subsequent paragraph.

The paper is organized as in sections; Introduction and problem definitions are in Sections 1 and 2. Section 3 illustrates solution overview. Section 4 has the knowledge abstractions and introduces two techniques *recapitulate* and *specification*. Section 5 discusses how the knowledge abstractions are used to reduce the size of the problem, the use of computation on *recapitulate.* In Section 6 we suggest a new heuristic method to divide the problem into subcomponents and classify, easy to solve and hard to solve (over-constrained) the subcomponents are interactive with a tree structure, these interaction explains the conflicts. Section 7 elaborates the conflict detection. In Section 8, we introduce an inadequacy of the disintegration strategy, namely *wide exam result*, and introduce to overcome the problem by defining *associations* among the conflicting problem components where the users can intervene in problem feasible solution and contribute preferences interactively in Section 9. We have applied the methods introduce earlier on to the framework to a real-life examination timetabling problem in Section 10. Section 11 draws conclusion and discussion.

## 2 Problem definition

Examination timetabling problem has to assign exams to periods (timeslots) and to rooms. This would be easy, except for the constraints that need to be valued as much as possible. E.g. a student cannot take 2 exams at the same time.

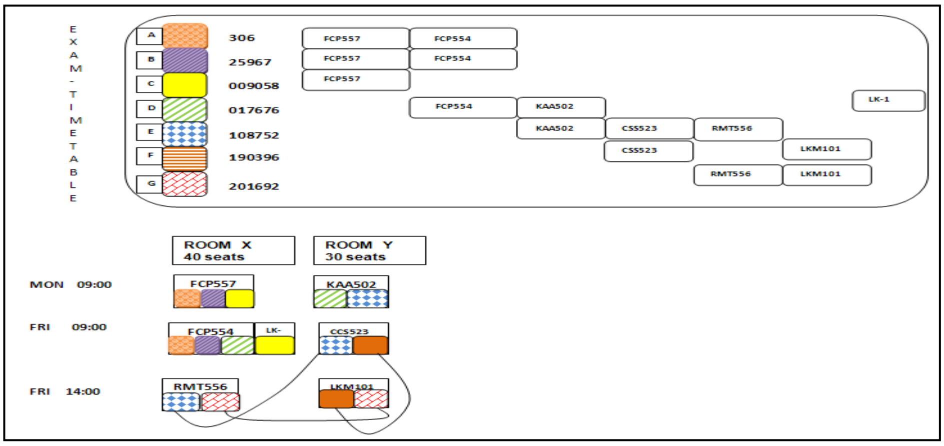



**Fig. 1.** Assignment of seven examinations to periods (timeslots) to rooms.

The above diagram illustrates an examination schedule in which students take which exams. For- example, 306 (student A) takes FCP557 and FCP554. But 25967 and 009058 also take the FCP557 exam. There are only 3 periods (Monday (AM), Friday (AM) and Friday (PM))) and 2 rooms (X with 40 seats and Y with 30 seats) available.

In a normal heuristic manner the algorithm orders and schedules exams to time periods and rooms based on the difficulty level, but it cannot guarantee the feasibility for all exams. 201692 (student G) has to take the RMT556 and LKM101 exams at the same time. And both 108752 and 201692 aren't too happy because they each have 2 exams on Friday.

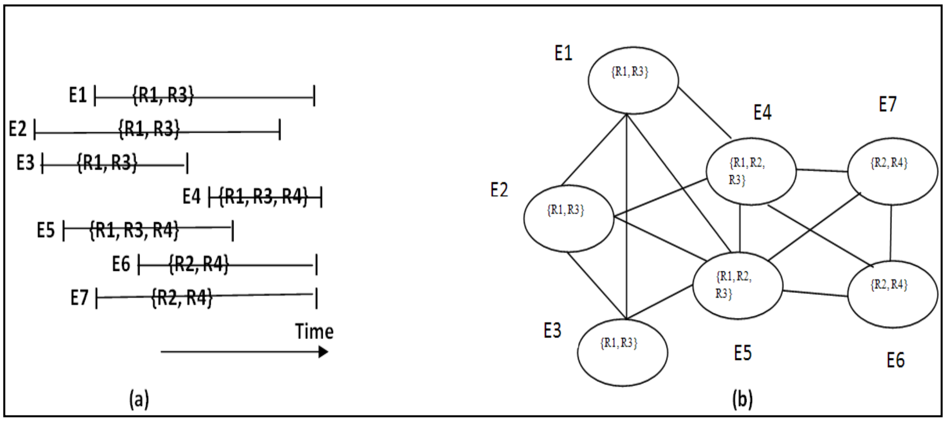



**Fig. 2.** (a) Examination Schedule of seven Exams with start time and duration during the scheduling phase. For each of the exams possible rooms are shown. (b) Constraint graph on the corresponding schedules.

The examination scheduling problem in Fig. 2(a) can be expressed as a discrete constraint satisfaction problem (CSP). To identify the CSP graph where the nodes are represented as exams to be assigned, Rooms are the resources, and arcs link nodes that intersect in time and indicate that one room cannot be assigned to more than one exam at a time. If it is the case, the constraint graph Fig. 2(b) generally highly connected and often has no feasible solution which can result as conflicts. Examination assignment can be easily mapped with a list coloring problem which is a category in graph coloring. The usual graph coloring problem in interval graph is known to be liner (Gupta et al. 1979) list coloring is NP-complete. (Arkin and Silverberg 1987).

## 3 Solution Overview

Examination timetabling has considered being NP-complete problem, to generate a solution method it is necessary to develop a structure graph which are more likely to fit the combinatorial problem. We suggest an architecture in which we use three useful techniques: *recapitulate, visual analysis heuristic (VAH)* and s*pecification.*

Recapitulate groups' successive constraints which can be executed by the same resource into a single one, thus reducing the size of the problem. This can be done when you look at the preprocessing of the problem (J.J Thomas et al. 2008).

In general, a *recapitulation* simplifies the problem and it might ignore feasible solutions. To avoid that we first steadily process the original problem to intermediate abstraction levels until confined *recapitulation* can be applied. The tree of splitting which reflect safe recapitulations, illustrated in Fig. 3. It explains the construction of the problem decomposition strategy called VAH (Visual analysis Heuristic). The problem division strategy iteratively decomposes the room allocation problem and expressed as in a CSP, (Constraint Satisfaction Problem) into tree structure consists of interacting sub-components. This clustering process, tests are carried out to check whether a *recapitulation* of the sub-component could yield a feasible solution, if not further refinement is applied. The box in Fig. 3 symbolizes the description of the diagram and its components.

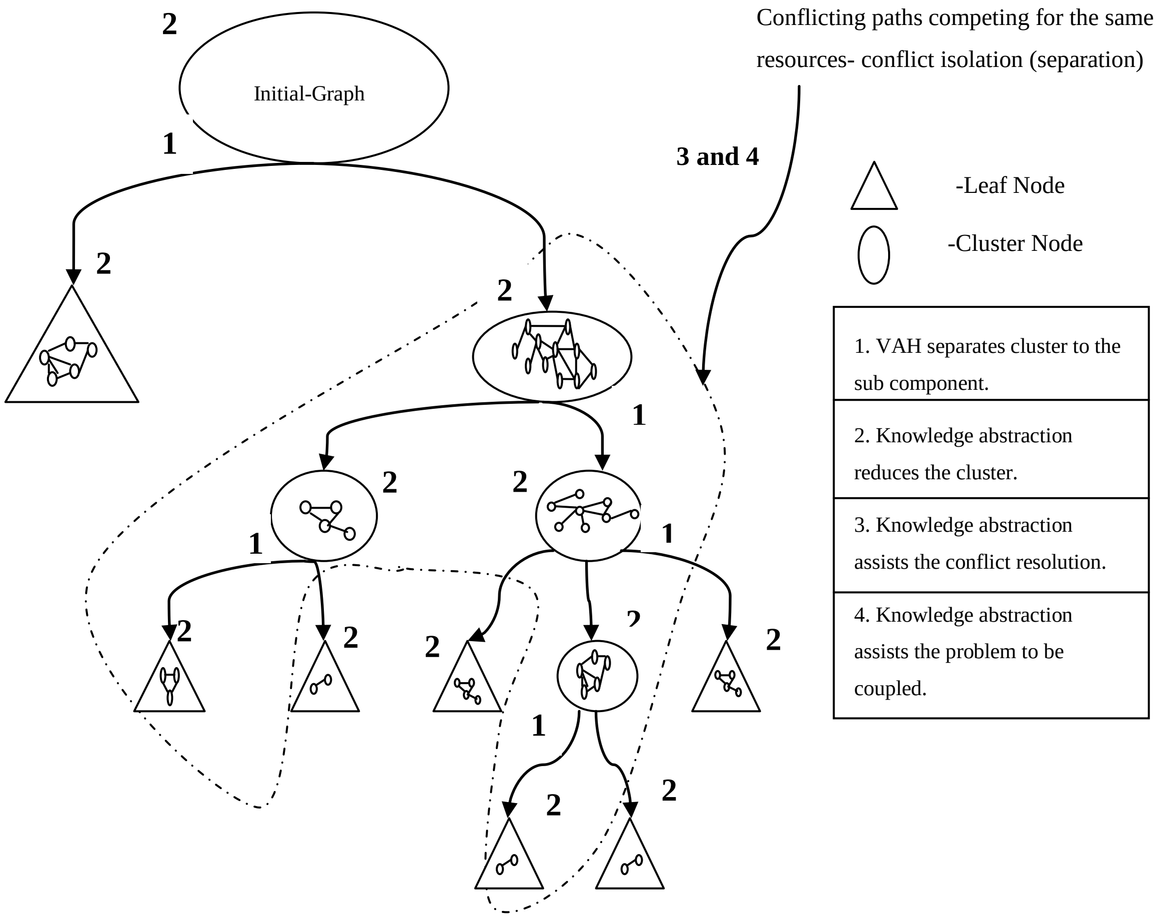



**Fig. 3.** VAH heuristic and knowledge abstractions in solving examination problem.

At the end of the clustering process, the leaves of the generated tree form isolated components of the initial problem. Some components are under-constrained and can be solved independently, for example room related constraints some other are over-constrained and interact among each other along the tree structure. There are interactions among the conflicts with the resource specific constraints as it treated as resource pools. It is located on the trunk of the tree. As shown in Fig. 3, the detachment strategy provides a framework (J.J. Thomas et al 2008), that is well suited for conflict isolation and for interactive problem solving, in which the human operator can intervene to select between conflicting paths in the clustering tree. This is very important feature, especially in heuristic application domains, for which preferences are difficult to formulize i.e. semi-automated schedulers.

A second knowledge abstraction technique named *specification* and is based on concept of specification (Dietterich, T.G. Michalski, J.C 1984). The process of specification generates abstractions bottom up along the tree structure in Fig. 3. It starting at the conflicting leaf nodes identified by the conflict resolution procedure. The two-level approach can easily extend into multiple levels to give constraint hierarchies (Boring et al. 1987). The specification procedure forms the most specific common specifications of the conflicting assignment. This is used as a feedback support for the visual analysis heuristics and evaluation of conflicts and problem solidity.

## 4 Knowledge Abstraction

Abstraction techniques have been proposed as promising methods to reduce the complexity of the problem solving and have been applied to large number of domains. In the examination timetabling problem, multi-user distributed environment with various cohorts of schools and department who often operate quite autonomously. It has been studied (Dimopoulou et al. 2001, Dietterich et al. 1984, Dean et al. 1987, Doyle 1987), much more work is required on understanding the issues involved and the interplay between user interaction and managing the information with the goal of producing a workable solution and the extent to which techniques can be used in an automated process.

Defining the abstraction is as follows: ( Golumbic 1984)

*Abstraction is the mapping of a problem representation into a simple one that satisfies some desirable properties in order to reduce the complexity of reasoning. The problem is solved in a simplified abstract space and the solution is then mapped back to the complex ground space.*

Abstraction techniques shows potential to reduce the complex problem, and it have been applied to wide number of domains. We suggest two processes of knowledge abstractions *recapitulate* and *specification.* Recapitulations group the knowledge space, and are considering only time related information (periods), while specifications group classes of similar within one proposition to form a more general solution. These methods are independent of adopted sequential formalism, we highlights the Time Map Manager (Dietterich et al 1984). The unit of TMM is sequential tokens, and it is associated with the event proposition of a time interval. In general abstractions are rough calculation and might lose little information related with the detailed descriptions of sequential token (constraints). We organize the tokens by using the arrangement used in TMS (Truth Maintenance System) ( Dean et al. 1987) the user has access to the sequential tokens (constraints) wherever it is necessary.

## 4.1 Recapitulate

Recapitulation is the substitute of a compilation of time tokens. The interval of the recapitulations is the smallest period that includes all intervals of the component tokens. The property of the recapitulation is a combination of the properties of the essential tokens, see Fig..4

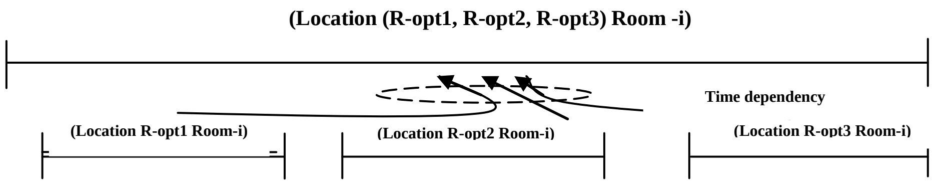



**Fig. 4.** Recapitulation of interval resource based constraints.

In the examination timetabling the allocation of rooms to exams are recapitulate such as opt1 takes place in room-1 followed by opt2 and opt3 in the same room (Institutional based soft constraint). All the examination must assigned to rooms within the time period (interval) during the exam week. The solution can be of constituent priorities. The priority does not necessary hold for every sub-interval. Some exams may need specific rooms, only exams of similar lengths are scheduled at the same timeslot in the same room.

After recapitulation, the abstract space comprises a subset of the initial set of possible solutions and any solution found in the abstract space can be safely mapped back to a concrete solution. In Fig.. 4 the assignment have been grouped to the same room, for alternative solutions. For example, assigning a different room for exams has been purposely ruled out and dropped from the solution space.

## 4.2 Specification

Specification is base on domain, depends on background knowledge in the form of concept hierarchy. A time table designer or administrator who has good experience in the hierarchy. For example an examination timetable designer must aware of what are the subjects are currently in the semester. To avoid clashes in exams, the basic knowledge is to allocate exams first for those who register for particular subjects in the current exams should be of fewer conflicts. Specification operates only on its propositional expression or property it provides a way to replace a disjunction of terms within one proposition, by a single more general term (Dimopoulou et al. 2001).

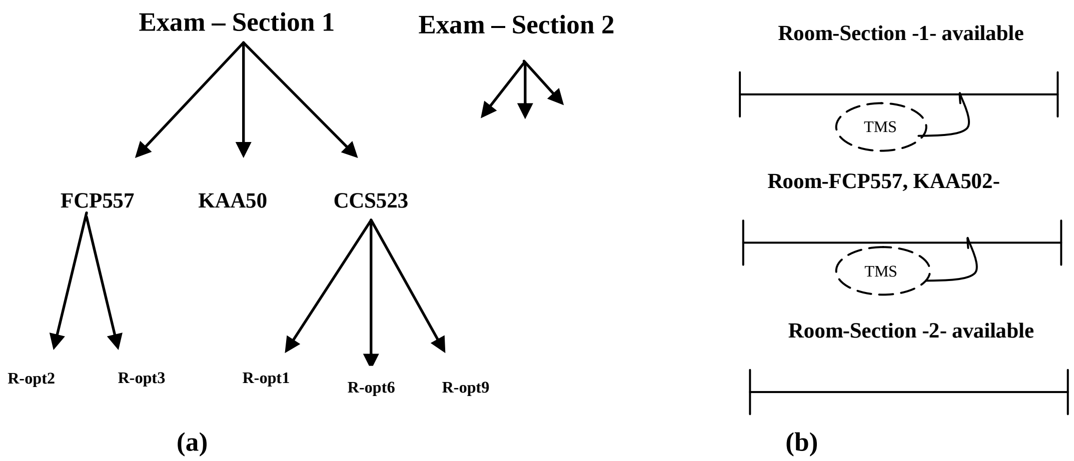



**Fig. 5.** (a) & (b) Specification using background knowledge structure

give another example of specifying a proposition obtained by recapitulation using specification hierarchy shown. For instance, the timetable designer recapitulates exam section -1, exam section-2 and so on and allocate to the timeslots (periods) into rooms with less conflicts. Specifications suppress the detailed descriptions of specific entities by mapping them to more generalized categories. Thus decisions made on the basis of the general categories alone might violate constraints that refer to the specific entities.

## 5 The use and computation of Recapitulate

The goal is to simplify the problem by reducing the number of assignment and thus the size of the search space, at the expense of reduced flexibility. In here, we discuss how recapitulation are applies in examination allocation. Each leaf cluster identified by the VAH is a set of assignments with similar room requirements. Recapitulations are applied to the leaf clusters in order to reduce their size. A leaf cluster “contains” a collection of intervals such as those shown in Fig. 6 (a). Groups of intervals which do not overlap are identified and replaced by exclusive interval. A regular allocation procedure can finally apply to assign one value per “time”.

The first step is to arrange the time interval, by topological sorting, in the directed graph reflecting their knowledge abstraction succession Fig. 6 (b). Nodes denote time internals, links are built between nodes that do not overlap, and arrows reflect their precedence order. Every path in this graph represents a possible sequence of assignment that can be carried out by one and the same room, and thus a possibly useful recapitulation.

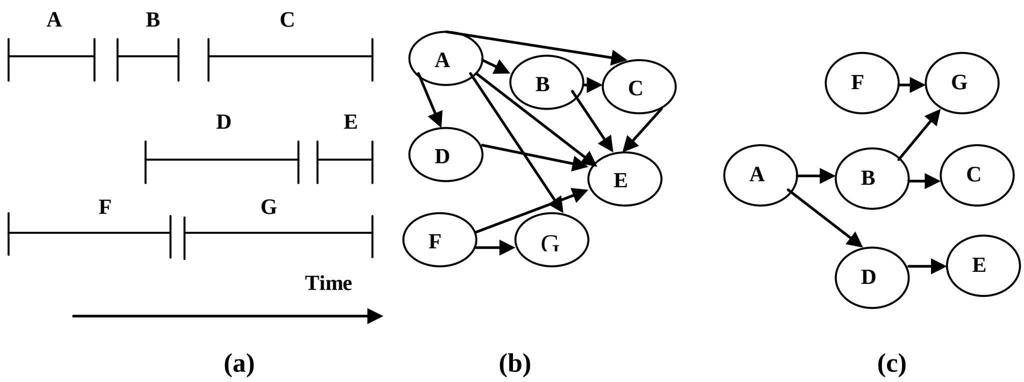



**Fig. 6.** (a) Token representation of timeslot located in one cluster. (b): graph comparability. (c): After eliminating arcs of the transitive priority.

The user may control the demanding constraints maximum, interval between correlated assignment and the arcs that exceed this interval are omitted from the graph. Competing recapitulations are apparent in the graph as different paths between the same nodes. Those that arise as a result of transitive priority are eliminated to simplify the graph.

If the users have some unquantifiable criteria (subjective preferences) for grouping assignments, they can be presented with the graph in Fig. 6 (c) so, they can interactively select the most suitable recapitulations. They can thus apply criteria which are difficult to formalize in a computer program, and obtain solutions which are more acceptable in practice.

The algorithm is to replace the set of intervals by the set recapitulated intervals of minimum cardinality. This can be done by coloring the intervals with the minimum number of colors then recapitulate those intervals with the same color. We consider the two criteria for establishing this order.

1. maximize the minimum distance between consecutive timeslot (allocating the exams, the length of the timetable)

2. minimize the maximum distance (the total number of students in the same room must be less than the capacity of the room).

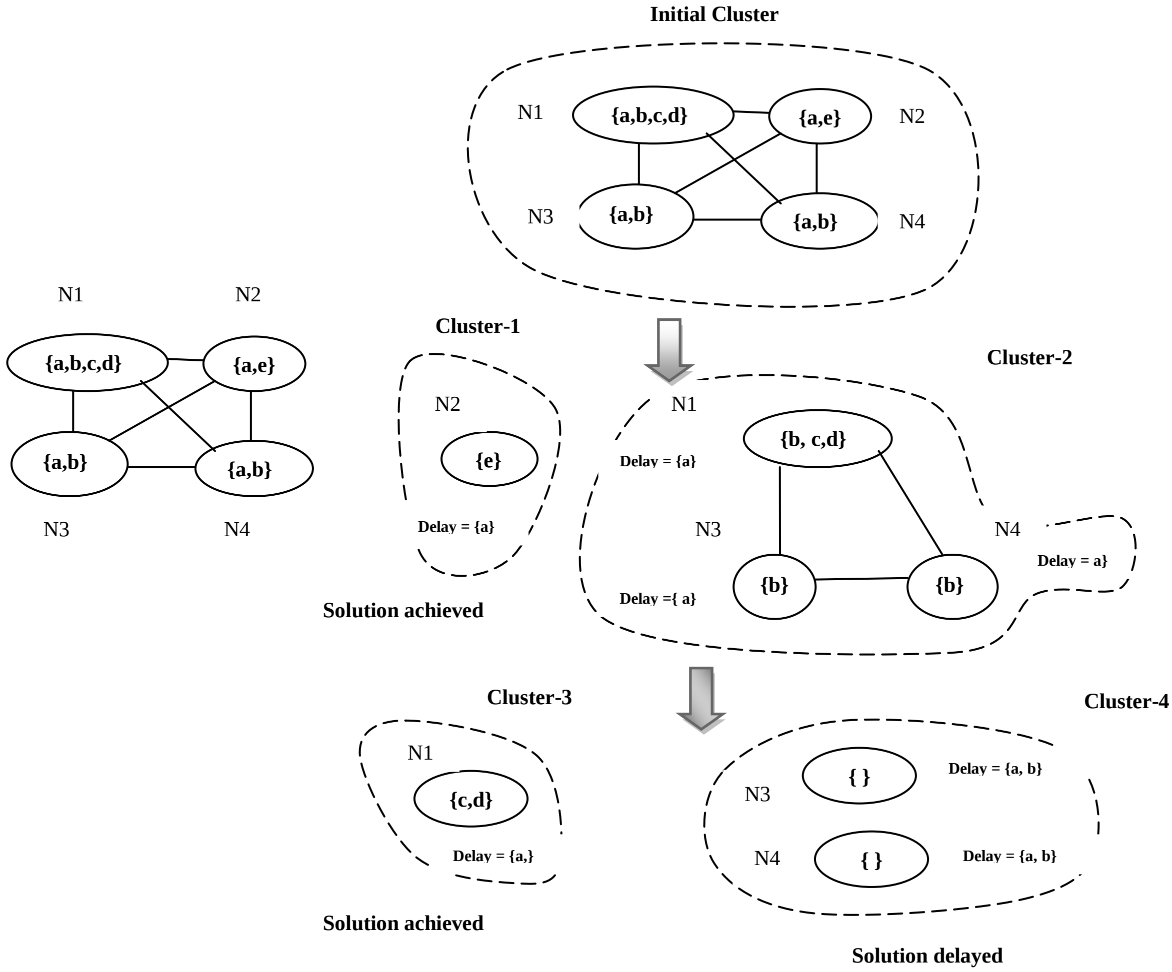
## 6 Visual Analysis Heuristic (VAH)

This section proposes a new heuristic called the Visual Analysis Heuristic (VAH) useful for solving list coloring problems expressed in Section 2 Fig 2(a) & (b). Here we are providing a node value called *delay*. This means that nodes linked cannot be given the same value. In interval graphs, two nodes are linked if and only if their time interval intersects and they have at least one value (i.e. resource) in common. Since, in real-life examination applications, many constraints are to be executed at the same time and many rooms are seek by many time slots, the corresponding constraint graphs tend to be densely connected.

The clustering process has happened in the nodes shared most common values in graph the elimination of arcs are applied to the constraint graph and step by step separating it to clusters. We demonstrate it with a simple illustration. In Fig. 7 (top), we show simple example of a list coloring problem with four nodes: N1, N2, N3 and N4. Each node has a set of values it can be assigned. It indicates the values assigned to the nodes are not the same. The backtracking procedure applies to the significant nodes and using iterations with respective of delay values to initial cluster and to the generated cluster in the bottom of Fig. 7.

Using the value-delay heuristic, the most common value, is delayed, thus splitting the initial cluster into two parts. In Cluster- 1, one can assign value {e} to N2 without hesitation. In Cluster2, the three nodes are still competing for common values, and they share, as a reserve, a set of delayed values, delay ={a}. At the next step, value {b} is delayed, and unsolved is split further, into two parts. One of these, Cluster3, can be assigned a value without further delay, and the remaining cluster is left with an empty set of possible values. It claims the delayed values “a” and “b”. A subsequent conflict resolution procedure assigns one delayed value to each unsolved node in this cluster.

The VAH heuristic provides a visually relevant, dynamically built, hierarchical structure to evaluate and it is an abstraction technique.




**(a)** **(b)**

**Fig. 7.** (a) List coloring problem (b): Applying Visual Analysis Heuristic to example

## 7 Conflict detection

Fig. 8 has three clusters namely Cluster 1, 3 and 4. Each cluster maintains values {a} and {b} and it can be solved independently. The root node has a delayed value {a} with two levels, the first level cluster (leaf cluster-1) are solved. Cluser-4 is claiming values {a} and {b} and the solution is delayed (not solved). Each set of delayed values maintain by one or more unsolved cluster called *conflict.*

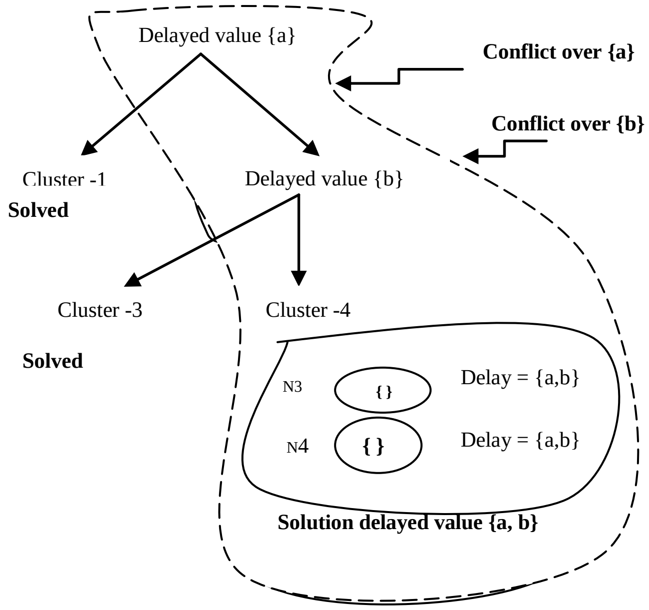


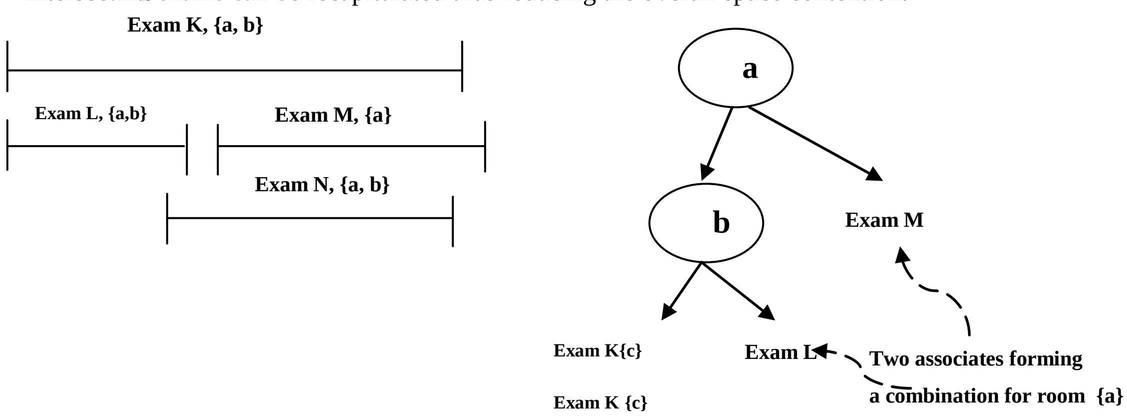

**Fig. 8.** Conflict Separation

The detection of conflict has been achieved by testing all conflicts by a conflict detection procedure. The conflicts are visually available for the timetable designers if it has preferences he or she can interact with the conflict resolution and modify the value assignment.

## 8 The Wide Exam Result and its resolution

The heuristics (or constructive) approaches are often stemming from a graph coloring heuristics. The basic timetabling problems can be modeled as graph coloring problem and the above section explains about the list coloring problem is a part of graph coloring problem. The conflict procedure clusters of a given branch node in the hierarchy tree might compete for values delayed however, the time interval does not interconnect. This will happen to all the clusters have competing same delayed values. We call this procedure as *wide exam result* it covers all instances of the clustering procedure.

When the exams in two apparently competing clusters, do not all intersect in time the two clusters defined a possible *association*. They can be merged and the non-intersecting exams can be recapitulated thus reducing the overall space contention.




(a) (b)

**Fig. 9.** (a): Four timeslots and their possible examinations. (b): illustration of Wide Exam Result method.

The example of Fig. 9 (a) is a very simple illustration of the wide exam result effect. Exam K has possible association with rooms {a, b} and Exam L with {a,b}, Exam M {a} and Exam N with{a,b} respectively. Fig. 9 (b) value {a} and {b} are the *association* forming the combination for room {a}. One way to overcome the wide exam result is to find possible grouping among leaf clusters. Upon user's request, the conflict detection procedure identifies all grouping and measures the corresponding decrease of exam conflict.

# 9 Automatic conflict resolution

Although our goal is to allow the users to selectively participate in the conflict resolution by viewing and manipulating conflicts and *associations* determined in the previous steps, we also provide an automatic conflict resolution procedure. This procedure may integrate domain dependent knowledge and adopt any of the following strategies:

## 9.1 Conflict resolution procedure

1. Allocate a delayed value to the first node encountered going in the reverse order of that used to create the tree (i.e. from the last visited leaf up to the root).
2. When two exams are competing for a delayed value, give it to the exam that temporally contains the other, or the one with the longest duration.
3. Allocate a delayed value to the exams that participate in the least number of associations.
4. Use domain dependent heuristic knowledge to distribute delayed resources over unsolved clusters. For instance, one may want to allocate a room to those exams that show closeness or distances.

These strategies are implemented with visual representation which will be depicted in Fig.10. Further investigation into conflict resolution is necessary. The conflict resolution procedure exhibits exponential complexity. However, since parts of the problem may have been solved by the previous procedure.

## 10 Real-world Examination Timetabling Problem: an example

In this section, we propose an *interactive visual model* for solving real-life examination timetabling problems that integrates the techniques introduced and discussed above. This solution method is made of three components, namely:

(1) a clustering algorithm based on the composition of the VAH heuristic and of knowledge abstraction.

(2) a conflict resolution procedure where conflicting needs of groups of tasks for the delayed values are identified and, either interactively or heuristically, solved, and

(3) a specification procedure carried out over unsatisfied constraints. This last procedure provides an assessment of problem rigidity. Here, we apply the solution method to a real-life examination timetabling as a case study: The allocation of examination to rooms based on standard constraints.

In this section, we demonstrates from Fig. 10 to 12, how a solution could be completed successfully. Firstly, a node consistency check is run to determine the rooms that can be allocated to each examination. Then the initial constraint graph is built and separated into independent connected graphs. Nodes that are not linked to the rest of the graph are isolated. Here we use three variables rooms, period and exams to visualize the exam to rooms and to period's conflicts and allocation. Fig. 10 has shown the possible conflicts, between rooms, examination and timeslot (periods).

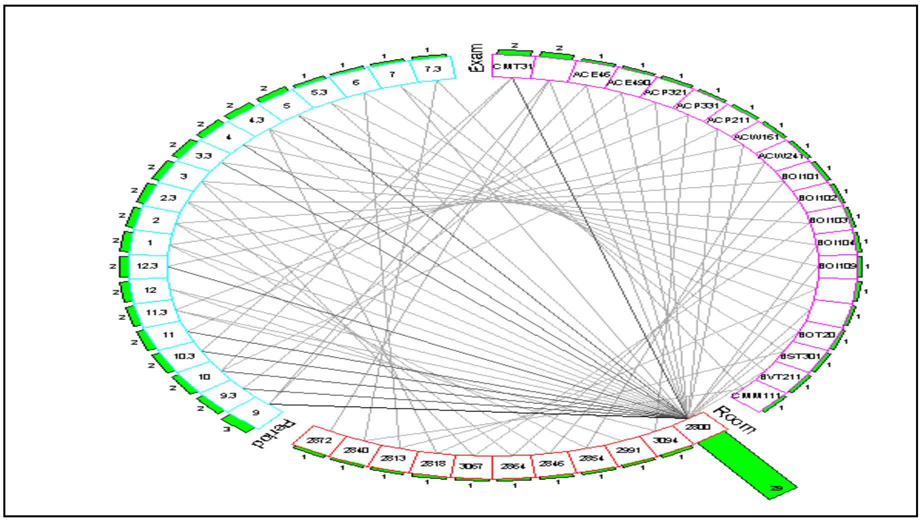



**Fig. 10.** The constraint graph of examination clashes between the exams to rooms and period.

The clustering procedure is applied to each of the remaining components. It uses the VAH heuristic, recapitulation; (Cheeseman et al. 1991)-Fig. 11 illustrated the initial problem discussed as in Fig. 3, the color symbolized the clustering groups and links between each examination. This visualization assists the problem to separate into two levels in later stage to apply the visual analysis heuristics techniques together with evolutionary algorithms.

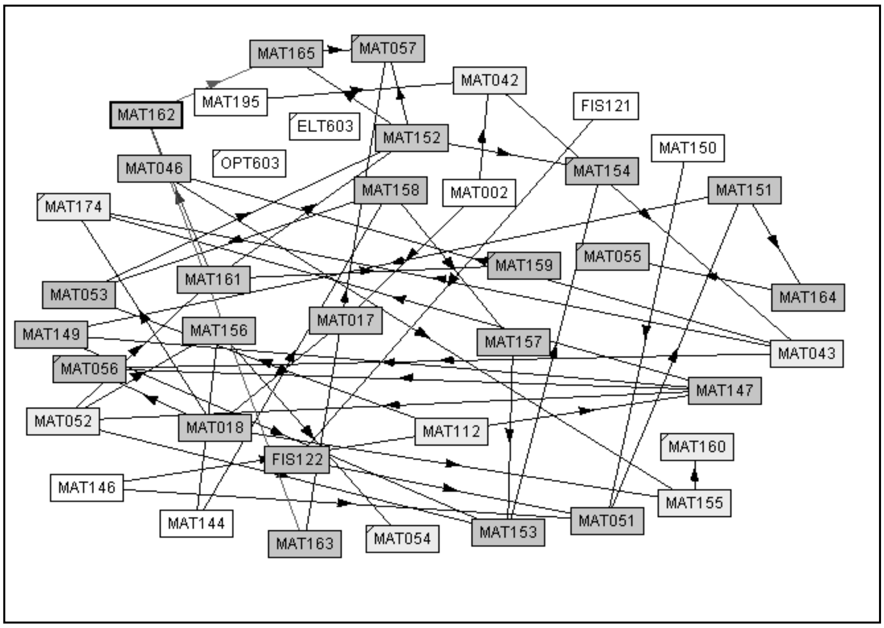



**Fig. 11.** Sample dataset color coded based on the exams clashes in a particular school.

The remaining in a leaf cluster (OPT603, ELT603) is solicited equally by the allocation of room in the cluster. The recapitulation process described is Section 4.1 is applied to these assignments to reduce the size of the corresponding leaf cluster. Fig. 12 shows the tree after the leaf clusters have been reduced by recapitulation; hence the clusters in this diagram are smaller than the corresponding ones in Fig. 11.

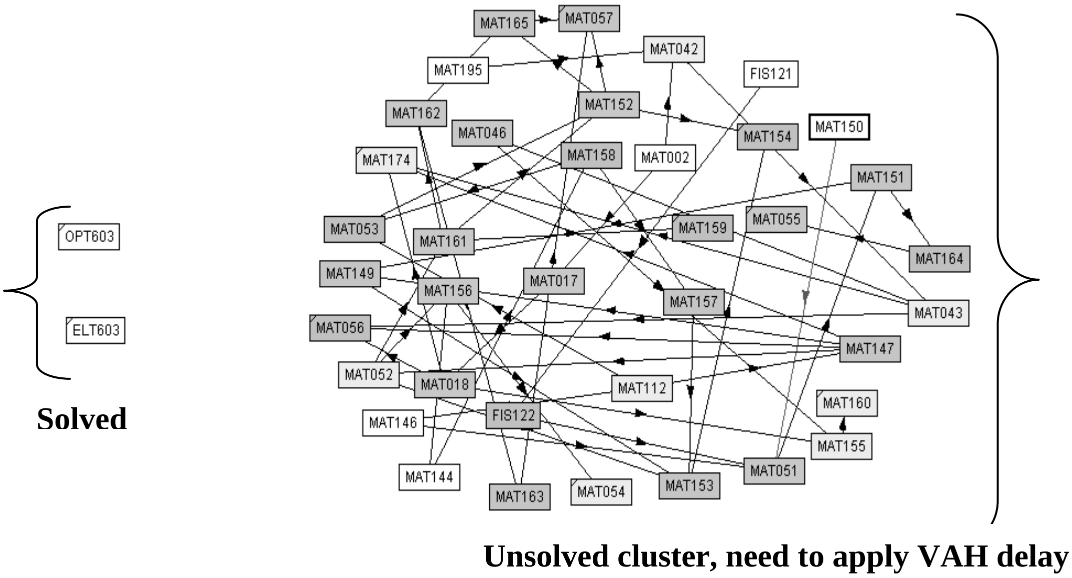



**Fig. 12.** Leaf nodes have been recapitulated and conflicts detected (Decomposition by VAH).

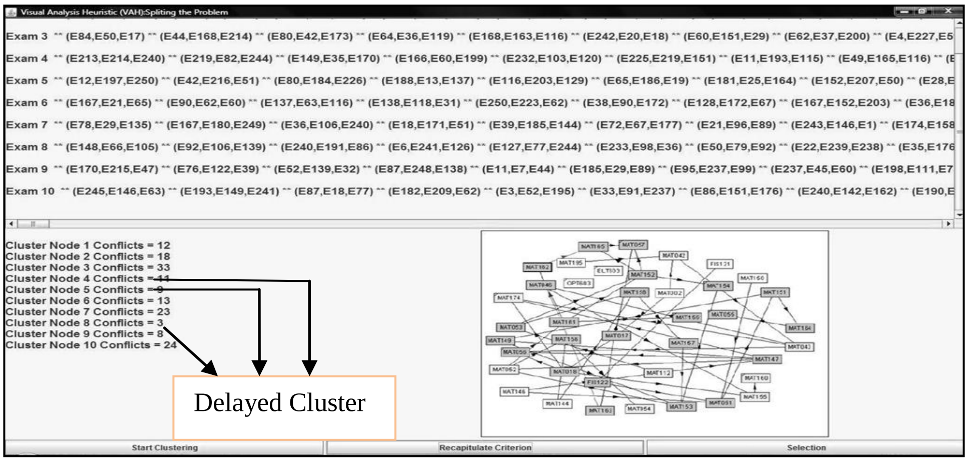



**Fig. 13.** Assignment tasks delayed values (Cluster 5, Cluster 8 and Cluster 4) using evolutionary algorithm

At this case, the user can see the current conflicts based on the delayed value assigned. For example here the delayed examination clusters are 5, 8, 4. The user may interact in the resolution of any conflict in allocation of examination to rooms.

However the solver is enable from the human scheduler by formation procedures and identifies the groups between the clusters claimed as delayed.

In Fig. 14. Visual Analysis Heuristic (VAH) solution split the problem into easily solved cluster and difficult solved clusters. By the method of *grouping* procedure identifies all possible clusters claiming the same delayed assignments. Finally, in Fig..14, the automatic conflict resolution succeeded in allocating exams to the rooms and all the grouped clusters are assigned with two assignments. E.g. Room No (1) assigns to Exam No (172), (107) same room used for two different examination in different period.

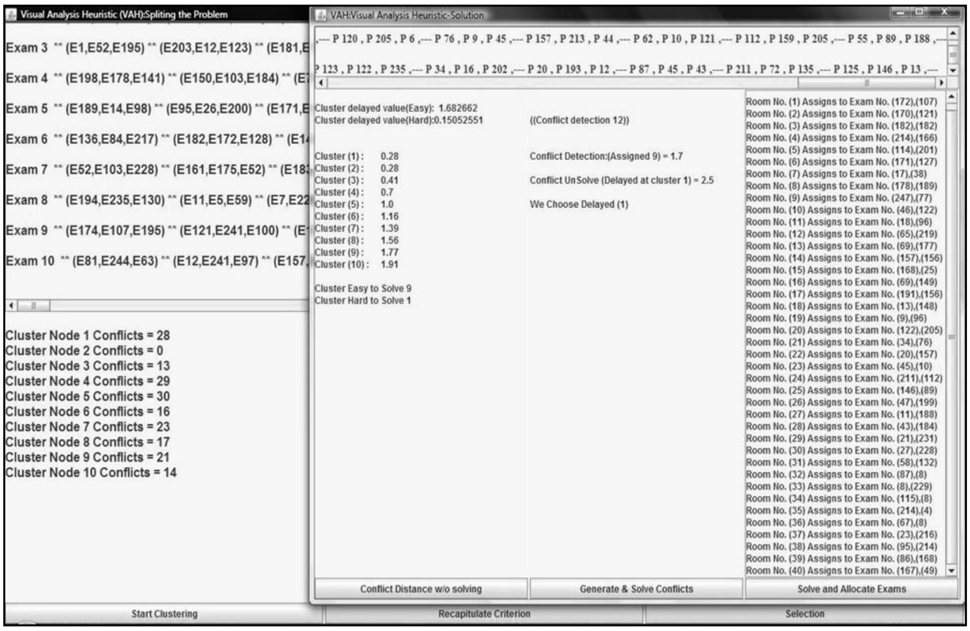



**Fig. 14.** Examination Assignment to rooms has been heuristically solved.

## 11 Conclusion and Discussion

We have presented a new interactive approach of the examination timetabling problem as it would appeal to many institutions. In providing this formulation, it is pointed out that minimizing the number of periods is not considered to be an effective way to solve the real-world examination timetabling problem. The discussion below is fruitful on helping the issues.

Several approaches have been used for solving the examination timetabling problems are based on variations to build an automatic solver. Many on backtrack search algorithms, mechanism are effectively implemented by underline the risks of combinatorial explorations. Majority of the times, it focuses on algorithm efficiency and the CPU time to solve the problem is discussed, importantly they fail to explain the reasons for the failure when it occurs.

Few remarks that have to be including in this paper, the proposed approach together with algorithms are creating a new breakthrough on the timetabling research. Interactive techniques and visual perception are work together to solve combinatorial problems with minimal violation of constraints especially to solve real-world timetabling application in general.

There are few approaches are highly suitable for interactive application *concept formation* rather than just writing on implementation with results. The is one of the good and advantages method is the reason provide to the reader it give visual pertinent hierarchical tree structures that give the user a closer walkthrough on the assignment of resources (real-world applications) and , through *recapitulation* and *specification* over conflict sets. It might assist the user to generate knowledge based abstraction for rescheduling.

Visual Analysis Heuristic (VAH) is assisted with delayed value to the clusters. Each group of clustering are formed is based on certain constraints (soft constraints) in real-world examination timetabling and with institutional model (Cumming et al. 2006) it is necessary to analyze the constraints and split to work towards a feasible solution. This technique is used for all type of list coloring problem which is a category of graph coloring problem always mapped to examination timetabling problem.

In many constraint based systems, contradictions are solved by relaxing one or more constraints choosing the minimum set of constraints known to be NP-Complete. But when we describe about feasibility and optimality, primarily the interested area is feasible solutions. Optimality is considered highly complex once the problem is in real-world environment (real-world examination timetabling). The clustering method we proposed will find a better near-optimal solution because we have avoided the situation where a few local minima are numerically compensated.

The prototype application has to be integrated to a single GUI and that will provide the user a greater flexibility. Furthermore, the benchmark dataset are to be tested to understand the proposed heuristics.

## 12 References


Al-Betar, Mohd., Khader, A.T., Gani, T,A.: A Harmony Search Algorithm for University Course Timetabling. In: Burke, E., Michel Gendreau (eds) Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling. Université de Montréal, August 18 - 22, 2008.

Arkin, E. Silverberg, E. Scheduling Jobs with Fixed Start and End Times. In.: Discrete Applied Mathematics, Vol 18, pp. 1-8(1987)

Burke, E.K., Kingston, J., de Werra, D.: In: Gross, J., Yellen, J.(eds) Applications to timetabling. Handbook of Graph Theory, pp. 445-474. Chapman Hall/CRC Press (2004)

Burke. E. K., Newall, J.P., Weare, R.F.: A Memetic Algorithm for University Exam Timetabling. In: Burke, E.K., Ross, P. (eds) Practice and Theory of Automated Timetabling. LNCS, vol. 1153, pp. 3-21. Springer, Heidelberg(1996)

Burke, E.K., McCollum, B., Meisels, A., Petrovic, S., Qu, R.: A graph-based hyper-heuristics for educational timetabling problems. European Journal of Operational Research 176, 177-192( 2007)

Burke, E.K., Marecek, Parkes, Andrew J. Rudova, Hana. Decomposition, reformulation, and diving in university course timetabling. Computers & Operations Research, Amsterdam, Elsevier, The Nederlands. ISSN 0305-0548, 2010, vol. 37, no. 3, pp. 582-597.(2010)

Benson, B. W. and Freuder, E.C. Interchangeability Preprocessing Can Improve Forward Checking Search. In.: Proceedings of the 10th ECAI, pp. 28-30, Vienna, Austria.

Berry, P.M. A Predictive Model for Satisfying Conflicting Objectives in Scheduling Problems. PhD thesis, Department of Computer Science, University of Strathclyde, Glasgow, UK.

Borning, A., Duisberg, R., Freeman-Benson, B., Kramer, A., Woolf, M. Constraint Hierarchies. In.: Proceedings of the conference on Object-Oriented Programming Systems, OOPSLA-87, pp. 48-60, Languages and Applications, Orlando, FL.

Bose, P. An Abstraction-based Search and Learning Approach for Effective Scheduling. In.: Famili, A., Nau, D.S., Kim, S.H (eds), Artificial Intelligence Applications to Manufacturing, pp. 187-197. AAAI-Press/The MIT-Press, Menlo Park, California.

Cheeseman, P. Kanefsky, B. Taylor, W.M. Where the Really Hard Problems Are. In.: Proceedings of the 12th IJCAI, pp. 331-337 Sydney, Australia(1991)

Cote, P.: A hyprid multi-objective evolutionary algorithm for the uncapacitated exam proximity problem. In: Burke, E.K., Trick, M.A.(eds) PATAT 2004. LNCS, vol. 3616, pp. 294-312.Springer, Heidelberg(2005)

Cumming, A. Paechter, B. Rankin, R.C. Post-Publication Timetabling, In: Proceedings of the 3rd International conference on the Practice and Theory of Automated timetabling, pp. 107-108 (2006).

Chiarandini M, Birattari M, Socha K, Rossi-Doria O. (2006) An effective hybrid algorithm for university course timetabling. J. of Scheduling 1094-6136 9: 403-432.

Di Gaspero, L., Schaerf, A.: Tabu search techniques for examination timetabling. In: Burke, E., Erben, W.(eds) PATAT 2000. LNCS, vol. 2079, pp. 104-117. Springer, Heidelberg (2001).

Dimopoulou, M. Miliotis, P. Implementing a University Course and Examination Timetabling System in a Distributed Environment, In: Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling. pp 148-151(2001).

Dietterich, T.G. Michalski, J.C. A Comparative Review of Selected Method for Learning from Examples. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M.,(eds) Machine Learning: An Artificial Intelligence Approach, pp. 41-81. Springer, Berlin, Heidelberg(1984)

Dean, T. McDermott, D. Temporal Data Base Management. Artificial Intelligence, Vol. 32, pp. 1-55 (1987)

Doyle, J. A Truth Maintenance System. Artificial Intelligence, Vol. 12, pp. 231-272 Morgan Kauffman San Francisco, CA, USA (1979)

Erben, W,: A grouping genetic algorithm for graph coloring and exam timetabling. In: Burke, E., Erben, W. (eds) PATAT 2000. LNCS, vol. 2079, pp. 132-158. Springer, Heidelberg(2001)

Eley, M.: Ant algorithm for the exam timetabling problem. In: Burke, E.K., Rudova, E.K.(eds) Practice and Theory of Automated Timetabling LNCS, vol. 1153, pp. 167-180. Springer, Heidelberg (1996)

Ellman, T. Abstraction via Approximate Symmetry. In.: Proceeding of the 13th IJCAI, International Joint Conference on Artificial Intelligence, pp.916-921, Chambery, France(1993)

Ellman, T. Synthesis of Abstraction Hierarchies for Constraint Satisfaction by Clustering Approximately Equivalent objects. In.: Tenth International Conference on Machine Learning, Amherst, MA (1993)

Feldman, R. Golumbic, M.C. Optimization Algorithms for Student Scheduling via Constraint Satisfiablilty. The Computer Journal, 3 No. 4:356-364.

Fox, M.S, Smith, S.F. ISIS: A Knowledge-Based System for Factory Scheduling. In.: Allen, J.F. Hendler, J. Tate, A (eds) Readings in Planning , pp. 336-360. Kaufmann, San Mateo, CA.

Gupta, U.J. Lee, D.T. Leung, J.Y. An Optimal Solution for the Channel Assignment Problem. IEEE Transactions on Computers, C 28 (11): 807-810 (1979)

Golumbic, M.C. Algorithmic Aspects of Perfect Graphs. In.: Annals of Discrete Mathematics. Vol 21, pp. 301-323(1984)

Giunchiglia, F. Walsh, T. A Theory of Abstraction. Technical Report IRST 9001-14, Istituto per la Ricerca Scientifica e Tecnologica, University of Trento, Trento, Italy (1990b).

Hart, E.Ross, P. Gavel - a new tool for genetic algorithm visualizationIn.: IEEE Transactions on Evolutionary Computing, Vol. 5, pp. 335-348.

Haselbock, A. Exploiting Interchangeabilities in Constraint Satisfaction Problems. In.: proceedings of the 13th International Joint Conference on Artificial Intelligence IJCAI, pp 282-287, Chambery France (1993)

J.J Thomas, Khader, A.T. Belaton, B. A Visual Analytics Framework for the Examination Timetabling Problem. In.: Proceeding of the 5th International Conference on Computer Graphics, Imaging and Visualization CGIV08, pp. 305-310, August 26-28 (2008).

J.J. Thomas, Khader, A.T. Belaton, B. Information Visualization Approach on the University Examination Timetabling Problem. In.: Book Visual Information Communication ISBN 978-1-4419-0311-2 pp. 255-264. Springer USA (2009).

Lewis R, Paechter B. (2005a) An empirical analysis of the grouping genetic algorithm: the timetabling case Evolutionary Computation, 2005. The 2005 IEEE Congress on, pp. 2856-2863 Vol. 2853.

Lewis R, Paechter B. (2005b) Application of the Grouping Genetic Algorithm to University Course Timetabling Evolutionary Computation in Combinatorial Optimization, pp. 144-153.

Lewis R, Paechter B, Rossi-Doria O. (2007) Metaheuristics for University Course Timetabling Evolutionary Scheduling, pp. 237-272.

McCollum B. (2007) A Perspective on Bridging the Gap Between Theory and Practice in University Timetabling Practice and Theory of Automated Timetabling VI, pp. 3-23.

McCollum. B, P.McMullan, E.K.Burke, A.J.Parkes, R.Qu, A New Model for Automated Examination Timetabling . In. Endre Boros (eds) Annals of Operational Research, 2008-accepted.

McCollum, B.: University timetabling: Bridging the gap between research and practice. In: Burke, E.K., Rudova, H (eds): Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling. 30th August – 1st September 2006, Brno, Czeh Republic, pp. 15-35 (2006)

McCollum B, Schaerf A, Paechter B, et al. (2009) Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition. INFORMS JOURNAL ON COMPUTING: ijoc.1090.0320.

Muller, T, Barak, R. Interactive Timetabling: Concepts, Techiniques and Practical Results, In: Proceedings of the 4th International Conference on the Practice and Theory of Automated timetabling, pp. 58-72.

Ow, P.S. Smith, S.F. Viewing Scheduling as an Opportunistic Problem-Solving Process, Annals of Operational Research, vol 12, pp 85-108. (1988)

Piechowiak, A. Ma, J. Mandiau, R. An Open Interactive timetabling Tool, In: Burke, E.K. Trick, M. (eds) selected papers from the 5th International conference, Pittsburgh 2004, Springer Lecture Notes in Computer Science, Vol 3616, Springer 2005.

Qu, R, Burke, E., McCollum, B., Merlot, L.T.G Lee, S.Y.: A survey of search methodologies and automated approaches for examination timetabling. Technical Report No. NOTTCS-TR-2006-4, School of Computer Science & IT, University of Nottingham (2006)

Rubio, R.G. A Timetable Production System Architecture for Course and Exams. In: Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling, pp. 342-350.