VIZSolution: An Interface Tool to *S*olve Real-world Examination Timetabling Problem

J. Joshua Thomas, Ahamad Tajudin Khader, Bahari Belaton, Amy Leow
School of Computer Sciences, KDU College Penang & University Sains Malaysia
joshopever@yahoo.com
doi:10.4156/ijact.vol2. issue5.9

Abstract

NP-Hard problems are hard to solve, it requires considerable computational power to generate solutions. The objective is to achieve better solution in less time. To obtain better solutions in semi-automated examination timetabling systems, we introduce visual interface tools to assist human to identify conflicts and turn the direction towards the feasible solution. We propose visual interface as a tool for the user to interact with the schedule generations.

Keywords: *Examination Timetabling, Graph Visualization, Visual Interface*

1. Introduction

The scheduling of examinations to time periods is a problem faced by many educational institutions at the end of the academic semester ([1], [12]). 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 [3], tabu search [10], evolutionary algorithms ([4], [2]) and simulated annealing [6], particle swarm algorithms [10], and harmony search algorithms [1] 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 ([14], [15]) 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 sot constraint violations [6]. 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 [12] was published. As such, the newer dataset released by ITC-2007 [18] 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 Interfaces, 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, during the processing and post processing ([13],[14]) over the examination timetabling problem.

There are a number of software commercial systems available for examination scheduling [12] 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 of rooms to timeslots and to exams. 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[5]. 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 Examination timetabling problem definitions are in Sections 1 and 2. Section 3 explains analysis of the expertise on the problem processes. Section 4 explains the design of ViZSolution system overview. Section 5 has the common architecture and illustrates the visual representation of the problem and the phase by phase solution techniques. Section 6 discusses how the solution has been attained with the guidance of visual interface i.e. ViZSolution. Section 7 draws conclusion and discussion.

2. Examination Timetabling

One of the most frequent application areas of informatics is scheduling. Scheduling problems come in all shapes and sizes. Our work concentrates on reactive schedulers as opposed to predictive ones. Given a global schedule and any related information, the reactive scheduler is responsible of accommodating any requested changes that may affect either the timetable of tasks or the allocation of resources to tasks [16]. Let's consider the university examination timetabling problem as the task.

Exam timetabling problems can be defined as assigning a set of exams into a limited number of timeslots (time periods) and rooms (of certain capacity), subject to a set of constraints. The complexities and the challenges presented by timetabling problems arise from the fact that a large variety of constraints, some of which contradict each other, need to be satisfied in different institutions [16]. In the timetabling literature, constraints are usually categorized into two types: hard constraints and soft constraints, which are explained below:

- *Hard Constraints* cannot be violated under any circumstances

For example, *conflicting* exams (i.e. those which involve common resources such as students) cannot be scheduled simultaneously. A timetable which satisfies all of the hard constraints is usually said to be *feasible*.

- *Soft Constraints* are desirable but are not absolutely critical. In practice, it is usually impossible to find feasible solutions that satisfy all of the soft constraints. Soft constraints vary (and sometimes conflict with each other) from one institution to another in terms of both the type and their importance [15].

The most common soft constraint in the exam timetabling literature is to spread conflicting exams as much as possible throughout the examination session so that students can have enough revision time between exams.

An example of another soft constraint which may conflict with this is to schedule all the large exams as early as possible to allow enough time for marking. The quality of timetables is usually measured by checking to what extent the soft constraints are violated in the solutions generated.

2.1. Primary Hard Constraints

- No exams with common resources (e.g. students) assigned simultaneously.
- Resources need to be sufficient (i.e. the number of students assigned to a room needs to be below the room capacity, enough rooms are needed for all of the exams).

2.2. Primary Soft Constraints

- Spread conflicting exams as evenly as possible, or not in *x* consecutive timeslots or days.
- Groups of exams required to take place at the same time, on the same day or at one location.
- Exams to be consecutive.
- Schedule all exams, or largest exams, as early as possible.
- An ordering (precedence) of exams needs to be satisfied.
- Limited number of students and/or exams in any timeslot.
- Time requirements (e.g. exams (not) to be in certain timeslots).
- Conflicting exams on the same day to be located nearby.

2.3. Timetabling languages and tools

Over the years, timetabling researchers have employed some general packages, such as ECLiPse for constraint logic programming [19], to build timetabling systems. However, some packages and languages which are specialized for timetabling have also appeared to support representations and comparisons in timetabling research.

This is a goal which (if it could be achieved) would undoubtedly benefit timetabling research. However, it is true to say that the various attempts to suggest such standards have not yet been built upon and widely adopted by the community [2], presented general requirements (generality, completeness and practicability) for building a standard data format for general timetabling problems based on set theory and logic. Examples were given to show how common constraints were modeled by using this data format. The objective is to provide an open way of making comparisons on results and exchanging data in timetabling research. There is no visual interactive tools are in the literature on solving the timetabling systems.

3. Analysis of the Expertise

The Examination scheduler presented in this paper is concerned with the allocation of rooms (resources) to exams and to the timeslot without major violation of constraints. The resource allocation problem in reactive scheduling is a hard one to solve. It is often bound to external, to the schedule, events that are hard to be anticipated in advance by the system. Even if the scheduler succeeds in taking into account all the needed constraints it may end up with an under-constrained problem that produces a large number of solutions, thus converting the initial combinational problem to a discrete optimization. We propose the visual interaction between the scheduler and the human skilled expert to involve in the problem solving process. There are few papers to blend interactivity with artificial intelligence techniques in order to solve the scheduling problems [22].

The user is actively involved into the process of problem solving and aids into the performance of the underlying computational scheduler. The communication link between the human expertise and scheduler must be bidirectional. The human expert at least should know the basic functionality of the algorithms to interact with the scheduler, in order to succeed in an effective interaction model.

For example, each of the solutions, if applied to the schedule, returns the schedule into a new symmetry. Even though human expertise is able to understand the process of the scheduler and the functions of the scheduling algorithms, it is hard to interact with the process. Instead, the system should respond with a comprehensive visual representation of all these solutions. By encoding in one display all of the parameter and constraints of the solutions are useful. We facilitate the processing of information thus leading the user to easier and more efficient to intervene the process of solutions.

In our work we propose a computer sustained framework for scheduling problem solving where interactive visualization enables a better mutual aid between the user and the machine (Scheduler). We consider interactive visualization as the main vehicle for seamless human-machine coordination in problem solving for rooms (resource) to examination allocation.

4. Presenting the system overview

"*ViZSolution*" is the environment allows human expert to collaborate and cooperate, sharing their cognitive with computational resources. It has enriched visual interfaces that assist expert has a benefit on understanding the process. In our case, the schedule parameters are visualized using parameters 2D graph visualization.

This human-centered design environment offers many software modules as servicing visualization shown as GUI in Fig.1. These visual representations assist humans define problems, parameter visualization, look for constraint violations and resolution, and discover the conflict process supported with visual interface tools and finally the scheduling solution.

Figure 1. ViZSolution Interface

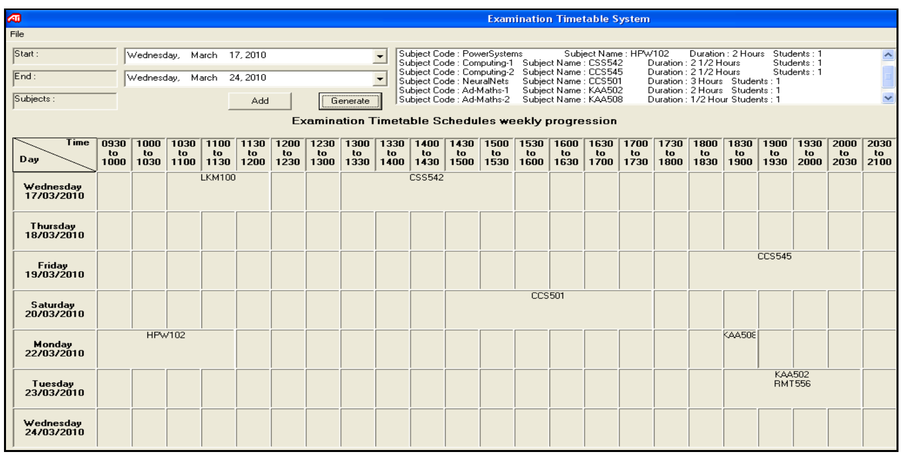

Figure 2. Examination Schedule Generations

There has been a significant amount of exam timetabling research in the last fifteen years. We note that many of the successful methodologies that have appeared in the present hybridizations of a number of techniques. Major research work has been done with search methodologies and techniques are in literature. There is no visual representation or techniques are addressed on the benchmark examination timetabling problem. To build visual interfacing tools to assist the scheduler is a complex task and it needs various artificial intelligence techniques are to be in-built to solve the problem. This complexity is hidden through a representational abstraction of the examination schedule in Figure 2.

In certain cases that dictate changes into the current allocation schedule. A good example is the overbooking/assigning of rooms. Since all of the rooms are all of the time assigned to specific exams the scheduling system has to shuffle some of the current assignments so that minimal conflict might happen and satisfied much soft

5. Design of ViZSolution

To assist the study of optimal problem solving systems we decompose the process of solving in three phases. In the beginning, the system constructs the necessary information workspace with which the user can interact. The goal of this stage is to better define the problem in itself and construct the appropriate input. The user can interact with the data; explore it looking for aspects of the problem instance. After the submission of the user's input the computational scheduler, the second phase on the search solution is activated. The search technique executes for solutions that satisfy both the users defined and the predefined constraints. The user can monitor the progress of the search or he can even interact with the scheduler in it.

Third stage is the problem solving process. Depending on the result the user may be presented with no feasible solution or multiple solutions. If no feasible solution has been found then the problem was over-constrained so the system and the user have to reevaluate the problem, find the violating constraints or the conflicts and relax them. When too many solutions are found, the problem is under-constrained and the user, with the assistance of the system, has to evaluate them and select the most appropriate one according to a set of constraints. On the other hand, they interact according to the needs of the problem solving process. For example, in a case of examination scheduling that we have highlighted, it is possible in the preprocessing phase to reach the optimal solution. This has been done with assistance of an interactive step by step processing of the problem space.

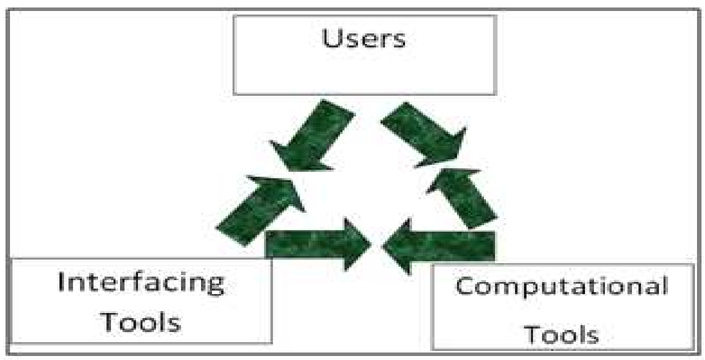

Figure 3. Interface System Design

It becomes apparent from studying the three problem solving phases that the computational tools and the human user have to work in a systematic throughout the process of problem solving. In order to achieve an effective co-operation there must be an efficient communication mechanism in place. In our experiment has showed us that interactive visualization may be the perfect way to reach a feasible solution with less time.

Machines are powerful for their computational skills and humans are particularly strong for their knowledge and creative qualities. In order to make them able to cooperate, we use a triangle design containing the user, computational tools and interfacing tools in Fig.3. Each item can communicate to another one. The user can benefit of the machine computational power and the machine can benefit of the expertise and of the intuition of the user. The interfacing tools provide as an interactive workspace allowing knowledge expert (human) and machine (scheduler) cooperation.

The user can use a same interfacing tool in order to communicate with several computational tools or use a same computational tool with different interfacing tools in order to have different points of view of a same problem solution. Furthermore, simple interactions between the computational tools and the users can take place in a straight forward manner without any mediators. However, complex interactive visualizations are more demanding and they require the presence of special interfacing tools. To make a bidirectional communication between the artificial intelligence techniques and human expertise a combination of interactive visualization tools are essential. It benefits the problem solving process.

5.1. Defining the problem

In order to define the problem, the knowledge expert is first using input as hard and soft constraints. The *Filter* assistant proposes the goal is to produce as many variations of constraints. Other assistance is provided in order to support the structure of the problem. The aim of the *Filter* assistant is both to keep a memory of initial conflicts, goals, and to support the formalization of the problem.

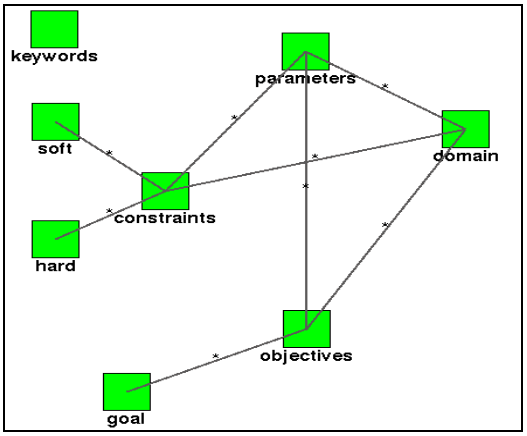

Figure 4. Interface System Design

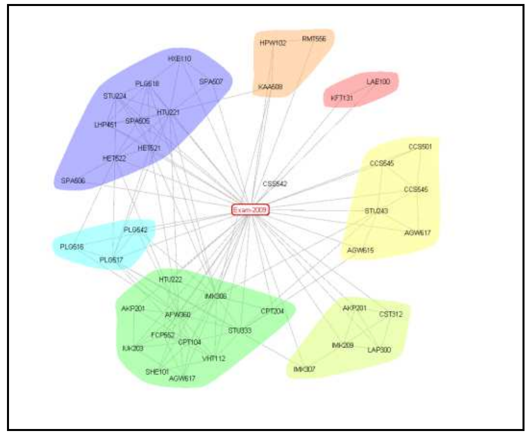

Figure 5. ViZSolution – Parameters FILTER-MGR

After defining constraints, the designer decides to use FILTER-MGR in order to define the problem in a more formal way and to be able to solve it (Fig.5). As the number of constraints increases, the resolution of the problem can get impossible for a human solver. Artificial intelligence techniques for search and for finding solutions in a constraints' network are well developed. We introduce the first visual representation based on clustering techniques to group the similar examination The definition follows the structure of a Constraint Satisfaction Problem. However, the constraints can be defined by either logic rules or matrices. Rules' writing can be a difficult task for a knowledge expert (human). The FILTER-MGR assistant supports this process by providing to the user a menu containing all the possible types of rule and their translation. It also provides a visualization of the graph created by parameters and constraints as shown in figure 3. It is a good visual feedback for the user because it gives a short view of the constraints' conflicts from the dataset.

5.2. In the Examination timetabling system

The available algorithms are attempted to satisfy the soft constraints, but not at the expense of violating any of the hard constraints. ViZSolution has a set of all the available information concerning the area of a scheduling application. They can also be found as part of the knowledge of the human operator. When a problem arises the operator needs to turn to the rescheduling application for reallocating resources. The system responds by building a visual workspace. The user can interact with it, adding or collapsing data from the information he draws from his knowledge. The final goal is to both refine the visual workspace and build a query that best represents the rescheduling problem.

Refining the workspace serves a double objective, it restricts the search space and it assists the user to understand all the constraint parameters of the problem and how possible solutions may affect the current status of allocations. In the case of examination scheduling each event in turn and places it into the first timeslot where no clash occurs and where capacity constraints are not exceeded. Extra timeslots are then opened for exams that cannot be placed into existing timeslots. It is obvious that a reshuffle has to take place so that another, larger in seating capacity, timeslot is assigned to the exams and at the same time the rest of the schedule can be performed in a similarly satisfactory way. Although the task in hand seems to be trivial, in reality it is a hard combinational problem.

First of all, the system is responsible of constructing and maintaining the visual workspace; a representation of a complex information of constraints, rules and interdependencies. Visualizing and interacting within this workspace is the only way to convey to the user all this amount of information. In our example the main visual abstraction being used in the visual workspace is the pictorial representation of the schedule, shown in Figure 2. The timeslots, depicted as rectangles, are ordered in the horizontal axis according to week day and in the vertical axis according to the timeslot currently assigned to them. Subject names, codes and time dependencies can be instantly seen.

It is important to note that everything the operator sees in his workspace specifies the subset of the whole schedule on which any changes can take place. It is only the exams, timeslots and time horizon being displayed that determines the boundaries of the search space. In the case of our upgrading example the user may build the visual workspace by selecting the exams to be reallocated and all the Rooms of the same type and of another type that has compatible characteristics but larger seating capacity. He can also restrict the time horizon in one week so that any proposed changes will affect the overall schedule within this time period only.

6. Solving the Problem

6.1. First Phase

We have implemented different algorithm for solving a CSP. We provide to the user visualizations of the going-through process of those algorithms. Thus in control of the searches and can decide to stop them whenever s/he wants. For example, a usual user interaction with our system consists in first using node consistency, and then arc-consistency. The user can then evaluate the number of solutions for the problem by using the *Knuth* algorithm. If it is under-constrained, he can either specify more constraints in order to compare the solutions. If it is over constrained, the user should use the elicit conflict agent in order to release the problem. In the case where the problem is neither over nor under-constrained, the user can do a backtracking. In visualizations are provided. It is a good way for the user to observe the processing and to control it. For example, the visualization in the figure 6 shows the going-through process of an intelligent backtracking, the numbers of conflicts are found while reaching the solutions are browsed. The idea is mainly to make the user able to use several kinds of algorithms, making them cooperate through interactive visualization.

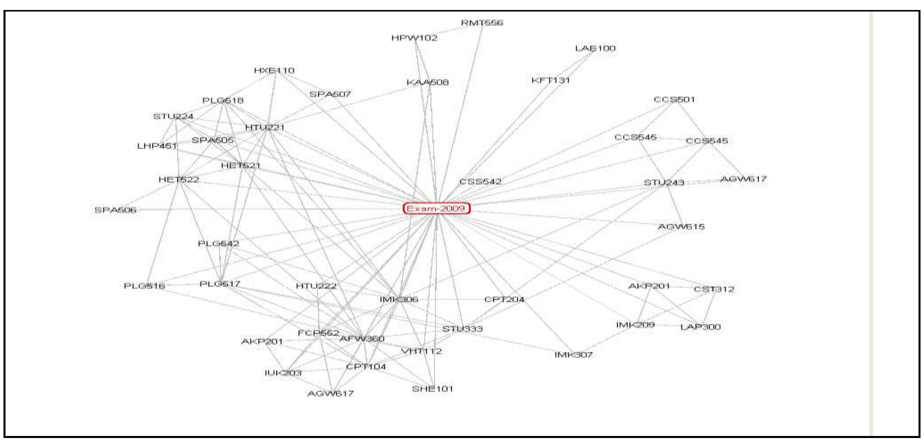

Figure 6. ViZSolution – Conflict Detection

6.2. Second Phase

After the user enters any additional constraints or preferences he invokes the system's scheduler. At this moment we pass into the second phase of the scheduling process. Although the examination scheduling application we have built offers limited interactivity and visualization during this particular phase, it is in some cases desirable to do so. When the search space is large and presents qualitative or quantitative regularities it is often advantageous to visualize both the process of searching and where the solutions are found. This permits to the user to either interrupt the search or continue with the already found solutions or to uncover possible areas of contention. According to the best case scenario the machine based search will respond with one and only solution that best meets the user query.

Unfortunately this is rarely the case in resource allocation. If the problem is over-constrained then no answer will be found. We can remedy this situation by returning to the visual workspace and relaxing some of the imposed constraints or simply enlarging its size.

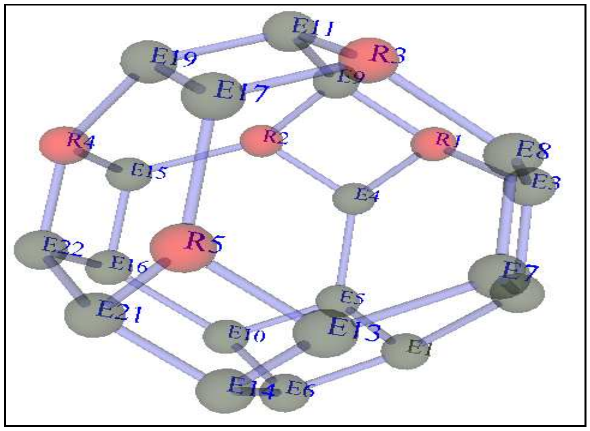

Figure 7. ViZSolution – 3D Conflict space

At this stage, the human expert must be engaged in a productive loop process of examining, evaluating and filtering out the solutions set. He can reason which solution best fits the problem, based on knowledge patterns (Fig. 7) drawn from experience, subjective preferences and/or additional criteria, such as the introduction of last minute qualitative constraints. The ultimate goal is to select the optimal solution.

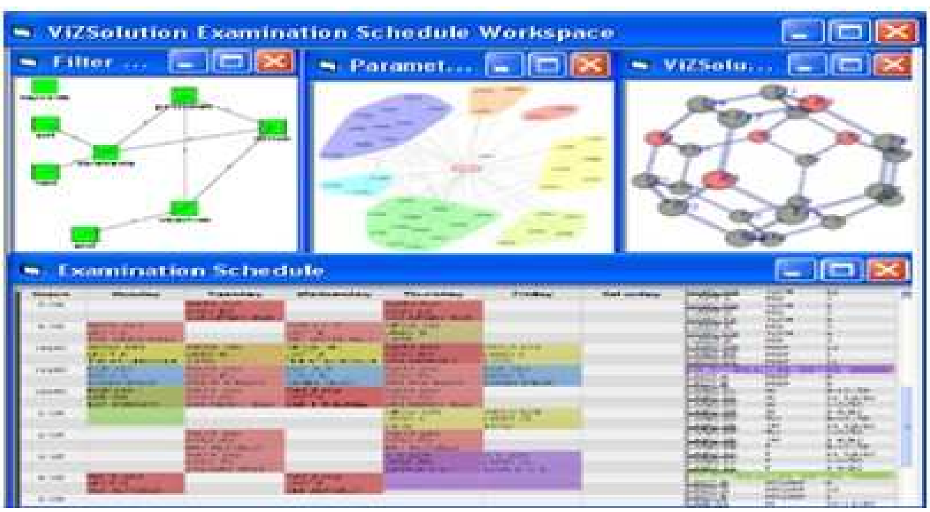

Figure 8. ViZSolution – Examination Schedule Visual Workspace

The user interacts, with any of the, displayed in parallel, representations in order to solve any perceptual ambiguities, deepen the level of displayed information and/or impose his own constraints thus limiting the number of available solutions. An intercommunication mechanism between the different representations creates an additional level of visual information flow towards the user. Therefore, the interrelations between solution attributes and contextual groupings of the solutions become apparent. All in all, information abstraction and interactive visualization along with machine based constraint satisfaction, free up the reasoning process by hiding unnecessary information details and speed up the ability to approximate the optimal solution.

In the examination scheduling system there are three morph types being utilized in the post processing phase, all displayed in the same screen, each occupying a quarter and in operational coordination with each other (fig. 8). The upper parameters portion of the interface is a map of the children nodes of a tree, build with the FDGL algorithm. The color children nodes represent all the available clusters in the solutions.

7. Conclusion and Discussion

We have experiment the complex problem solving with visual interface tools on the examination timetabling problem. The system design can be applied to many NP-hard problems where the artificial intelligence system fails; the human intelligence can be used as a stimulator for solving the bottleneck situations. This work has been in progress, and more work has to entangle to improvise the solutions.

8. References

[1] 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)

[2] 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)

[3] 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)

[4] 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)

[5] 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.

[6] 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.

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

[8] 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)

[9] 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)

[10] 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)

[11] 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)

[12] 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.

[13] 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.