An Intelligent, Interactive & Efficient Exam Scheduling System (IIEESS v1.0)

Zhu Chunbao and Tha Nu

School of Engineering, Nanyang Polytechnic, Singapore
Tel: (65) 65501808, Fax: (65) 6454 9871
Email: ZHU_Chun_Bao@nyp.gov.sg

Abstract The purpose of this paper is to introduce and demonstrate an exam scheduling software system that performs efficient, accurate and robust solution searching to solve large and complex exam scheduling problems. The paper describes the main features of the system and in particular the test paper conflictive analysis method that can provide a highly efficient data model to significantly improve search efficiency and interactivity.

Keywords Exam timetable scheduling, test paper conflictive analysis, swarm intelligence, indirect clash-checking

1 Introduction

The IIEESS v1.0 is an intelligent, interactive & efficient exam scheduling system designed and developed to solve large and complex exam scheduling problems using the patent pending technologies (Zhu Chunbao, 2008, 2010). The software solution can be applied to schools, institutions, universities and training centers which need to schedule examination activities and allocate venue resources to facilitate these activities.

Traditional exam scheduling systems directly examine the vast amount of student registration data for checking student conflicts and constraint violations repeatedly during solution searching cycles. This is not efficient and not robust particularly when iteration based search algorithms are used, such as GA and ACO based systems (Shu-Chuan Chu, Yi-Tin Chen, Jiun-Huei Ho, 2006). As results, computer runtime is lengthy (Nelishia Pillay and Wolfgang Banzhaf, 2007), such as for example, it takes 4 ~ 5 hours for the program to search for a solution.

Unlike direct clash checking, the IIEESS v.10 system firstly carries out the test paper conflictive analysis, which yields a conflictive coefficient matrix $\Omega_{n\times n}$ and then further creates the mutually exclusive paper lists MEL_k (k=0 to n, n is the number of exam papers). Note that the number of elements in MEL_k, denoted N_{MELk} is much less n. The system then indirectly examines the conflictive coefficients in MEL_k for student conflicts in solution searching cycles, rather than directly examining the huge amount student registration records and original constraints imposed. The system also utilizes the conflictive coefficients to minimize constraint violations to further increase the system's efficiency and accuracy.

Because the number of exam papers n (say hundreds) and N_{MELk} (say tens) is much less than the number of student registration records (tens thousands to millions), the new system enjoys high efficiency, accuracy and robustness. Our computational experiments show that the IIEESS v1.0 system is much faster than direct-clash-checking systems. The high efficiency enables the new system not only to provide fast auto-searching, but also to facilitate the system with user-friendly and truly effective drag-&-drop features which are critically important for planners to perform manual amendments to the auto-generated schedule.

The system provides a powerful automatic venue resource allocation engine and user-friendly drag-&-drop features as well for facilitating the scheduled examinations.

2 The Method of Test Paper Conflictive Analysis

To simplify the explanation of the paper conflictive analysis, Fig. 1 utilizes two test papers, $Paper_i$ and $Paper_j$ for illustration purpose. The set of students, who take $Paper_i$ and $Paper_j$ denoted S_{Pi} and S_{Pj} respectively. The intersection of the set S_{Pi} and the set S_{Pj}, is denoted as $\Delta S_{Pi,Pj}$.

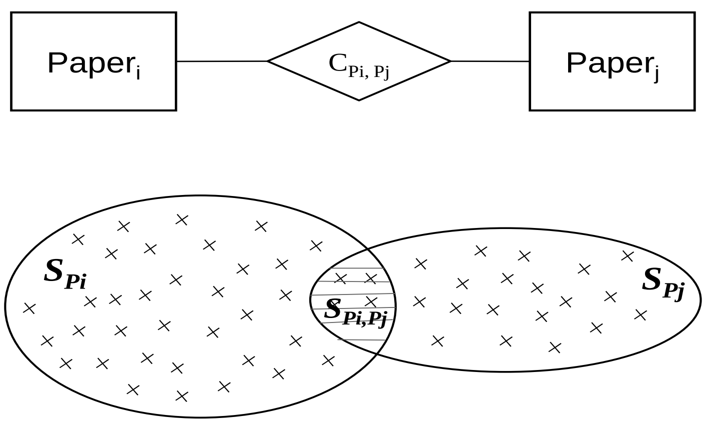


Fig. 1 Test paper conflictive coefficient

It is important to note that the candidates, if any, in the intersection of the Set S_{Pi} and Set S_{Pj} are *common students* who take both $Paper_i$ and $Paper_j$. If $\Delta S_{Pi,Pj}$ is not empty, i.e., the number of students in the intersection set $\Delta S_{Pi,Pj}$ is large than zero, we conclude that $Paper_i$ conflicts with $Paper_j$. By conflicts, we mean that the $Paper_i$ and $Paper_j$ cannot be scheduled at same period of time because they have at least one common student. In other words, $Paper_i$ and $Paper_j$ are mutually exclusive each other.

Indirect Constraint Evaluation Method

In scheduling, it is convenient for the search engine (program) to examine the original constraints imposed to avoid constraint violation, which is "direct-constraint-checking". Indirect constraint evaluation method performs the task in two separate steps. First step is to study the event conflictive features among all events according to the original constraints imposed. Output of the first step is a short checking list. The second step is to check the short checking list rather checking the original constraints imposed to avoid constraint violation in solution searching cycles.

Note that the first step is one time operation before solution searching; the second step is to be repeated in solution searching cycles. Because generally for large and complex ETPs, the short checking list is much shorter than the original constraints imposed, therefore the indirect checking method enjoys much higher efficiency, accuracy and robustness. In this paper, the short list used to check student conflicts is constructed using the test paper conflictive coefficients which will be described as follows.

The test paper conflictive coefficient

It is convenient to use a single numerical number (integer) to describe the conflictive relation among test papers. We utilize the paper conflictive coefficient, $C_{Pi,Pj}$ to measure how two test papers are conflictive each other, where the indexes P_i and P_j refer to any two test papers in the schedule. For example, $C_{Pi,Pj}$ is the paper conflictive coefficient for $Paper_i$ and $Paper_i$ as shown in Fig. 1. In general, for every two papers, $Paper_i$ and $Paper_j$, the paper conflictive coefficient, $C_{Pi,Pj}$ can be obtained as follows: $C_{Pi,Pj} = |\Delta S_{PiPj}|$. Where S_{PiPj} is the intersection of the student set for $Paper_i$ and $Paper_j$; and $|\Delta S_{PiPj}|$ denotes the cardinality of the intersection set ΔS_{PiPj}.

Fig. 2 shows that the exam events can be more efficiently scheduled using test paper conflictive coefficients.

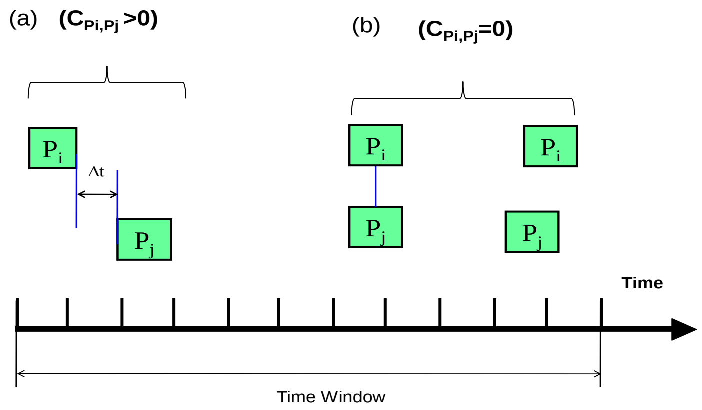


Fig. 2 Exam event scheduling using test paper conflictive coefficients

For example, if $C_{Pi,Pj}$ is equal to zero, $Paper_i$ and $Paper_j$ are independent each other; which means that they can be scheduled at same time slots or at a different time slot but with overlapped period, as can be seen in Fig. 2 (b); otherwise $Paper_i$ and $Paper_j$ are mutually exclusive, which means that they cannot be scheduled at same time, there must be a time gap ($\Delta t>0$) between the two exams, shown in Fig. 2 (a).

The quantitative value of the paper conflictive coefficient, $C_{Pi,Pj,}$ is important for the system to evaluate soft constraint violations. For example, if two mutually exclusive exam papers, such as $Paper_i$ and $Paper_j$, are scheduled with a narrow time gap (Δt), which will result in B2B constraint violation, or "Multiple Exams A Day Conflicts" - the multiple papers are scheduled on the same day, the system has to examine $C_{Pi,Pj}$ which is the number of students involved. It is necessary to minimize the total number of students who are scheduled to do multiple papers within one day.

Fig. 1 and Fig. 2 show two test papers for illustration on how to use a paper conflictive coefficient to measure the conflictive grade. In practice, the number of test papers, denoted n, can be quite large. Therefore, it is necessary to express the conflictive relations among *n* papers, that is, P_1, P_2, … P_{n-1}, P_n; the matrix of the conflictive coefficients among n papers is introduced, its denotation is Φ. Where Φ is an $n\times n$ matrix expressed as $\Phi = [C_{i,j}]_{n\times n}$. Where, the element $C_{i,j}$ is the conflictive coefficient for $Paper_i$ and $Paper_j$. Let $C_{i,j} = 0$ if i=j; because a paper can never be conflictive or mutually exclusive with the paper itself. It is noted that Matrix Φ is symmetrical, that is, element $C_{i,j} = C_{j,i}$ because conflictive nature between $Paper_i$ and $Paper_j$ is same as the one between $Paper_j$ and $Paper_i$.

The paper's mutually exclusive paper lists

Furthermore, we remove the elements whose value is zero from in Matrix Φ, we can get a shorter conflictive coefficient list and then obtain a mutually exclusive paper list for every paper as shown in Fig. 3.

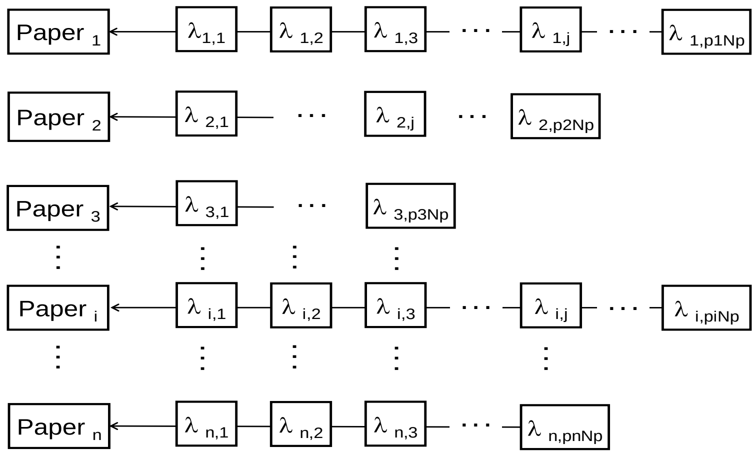


Fig. 3 The mutually exclusive paper lists

For example, for $Paper_i$, its mutually exclusive paper list is represented using a list of value pairs, each being denoted $\lambda_{Pi,Pj}$ as $\lambda_{Pi,Pj} = \langle P_j : C_{Pi,Pj} \rangle$.

The element $\lambda_{Pi,Pj}$ is called <paper index - conflictive coefficient> value pair. Fig. 3 shows the mutually exclusive paper lists in form of the value pairs, $\lambda_{Pi,Pj}$. Note the subscript, i = 1 to n, where n is the number of total test papers; j = 1 to p_iN_p, where p_iN_p is the total number of conflictive papers which are conflictive with Paper P_i. As can be seen from Fig. 3, the value of p_iN_p varies. In general it is much less than n in most of the cases. An exemplary value of the value pair $\lambda_{P10,\,P20}$ is $\langle P_{20}$:400$\rangle$, which means that paper P_{10} is conflictive with the paper P_{20}, and the number of students who take both paper P_{10} and paper P_{20} is 400.

As can be seen from Fig. 3, the value pair $\lambda_{Pi,Pj}$ is used to construct the mutually exclusive paper lists. For paper P_i, its mutually exclusive paper list MEL_i is expressed as follows.

$$MEL_i = \{ \lambda_{i,1}, \lambda_{i,2}, \lambda_{i,3}, \ldots \lambda_{i,j} \ldots \lambda_{i,PiNp-1}, \lambda_{i,Pi\,Np} \} \qquad (1)$$

It is the mutually exclusive paper list MEL_i that is used in the IIEES 1.0 system for clash-checking. That is, if $T(Paper_i)$ denotes the time slot scheduled for $Paper_i$, $T(Paper_j)$ denotes the time slot scheduled for $Paper_j$, for any $Paper_j$ which is in $Paper_i$'s mutually exclusive paper list MEL_i, following student conflict free constraint must be satisfied.

$$T(Paper_i) \neq T(Paper_j) \qquad (2)$$

If the number of students taking paper P_i, is P_iN_p, say 500 students, that is, who were enrolled with $Paper_i$ as shown in Fig. 1, and the number of students taking $Paper_j$ is 400, traditional direct clash checking method has to make massive comparisons 500×400 = 200,000 in order to find if there is any student conflict. However, using the new indirect clash checking method, the system only needs to check whether the $Paper_j$ is in $Paper_i$'s mutually exclusive paper list MEL_i, if answer is yes, $Paper_j$ and $Paper_i$ cannot be scheduled at same time due to student conflicts.

It should be highlighted that to check student conflicts using the indirect clash checking method, the number of comparisons is P_iN_p, which is the length of $Paper_i$'s mutually exclusive paper list MEL_i. Typical value of P_iN_p for a medium sized exam scheduling problem, is tens, say 0 to 20, which is much less than $P_iN_p \times P_jN_p$, say 200,000 as described previously. As the result, the new indirect constraint checking method is many times (i.e., 10,000) faster than traditional direct constraint checking for the example illustrated in Fig. 1.

3 The System Architecture and Software Modules

The IIEESS 1.0 system consists of three functional modules: 1.) input module, 2.) exam scheduler, and 3.) reporting module, as shown in Fig. 4. The input module consists of a data loader that down-loads student registration data from external sources such as databases or other forms of data storage. The input module also stores into the internal storage the exam scheduling information such as exam papers, venue facilities, and time slots as well as constraint information.

Once registration data are loaded into the system, the test paper conflictive analyzer will performs data pre-processing, test paper conflictive analysis in particular, and store the paper conflictive information for the exam scheduler to use. The scheduler contains an internal storage, a timetable scheduler and venue resource allocation module. The reporting module provides functions to upload the exam schedule solution to legacy database such as exam management system, and it can also generate various exam timetable reports for trail release or formal publication.

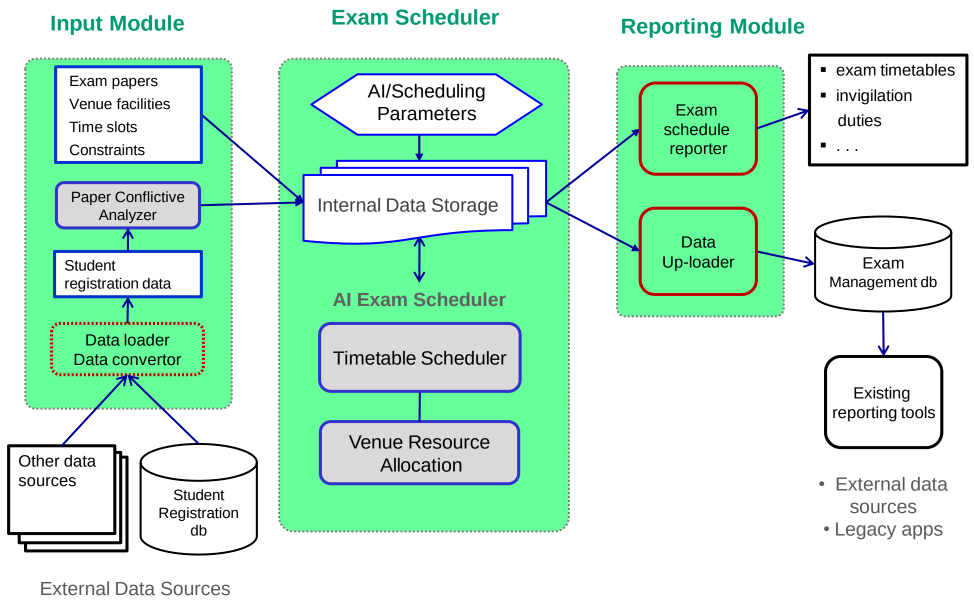


Fig. 4 IIEESS 1.0 System Architecture

4 Input Data

The input data to the IIEESS v1.0 system are categorized as follows: 1.) student registration data, and 2.) exam scheduling information. The student registration data describe “who studies what and in which group?”, i.e., the candidates-papers relationships, which is critically important because the exam scheduler has to generate a conflict-free timetable solution. The exam scheduling information includes the following:

1.) Exam paper information and related constraints
2.) Venue facilities and capacity/availability constraints
3.) Time slot specification
4.) Soft constraints

Student registration data

The student registration data can be presented and stored in many different forms. The format adopted by the IIEESS 1.0 is as follows.

<Student Admin No> <Module Code> <Module Group>

The school name and campus code in Fig. 5 are specific in our school and for venue resource allocation use. Note that the student admin number, module code and module group must be unique in the schedule.

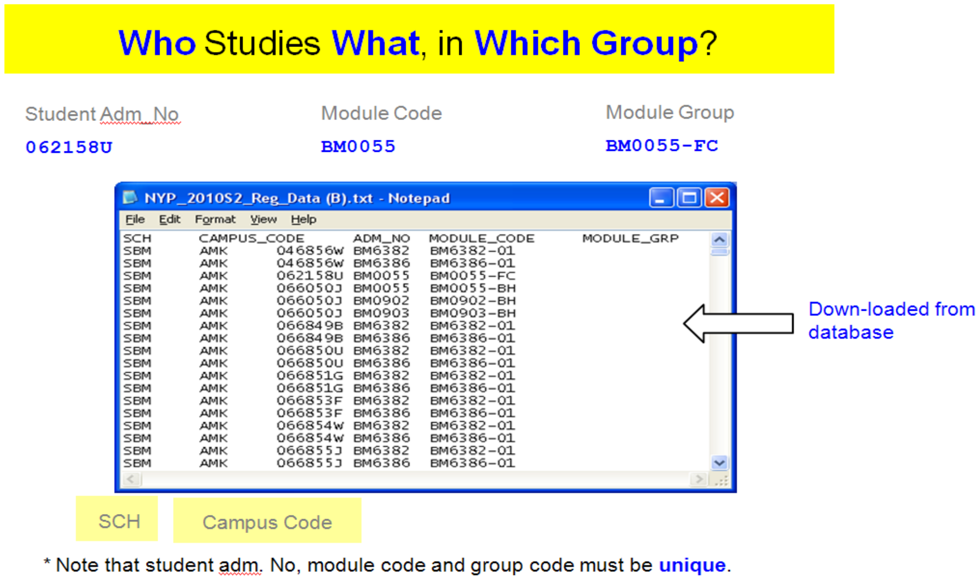


Fig. 5 Student Registration Data

Time Slot Specification

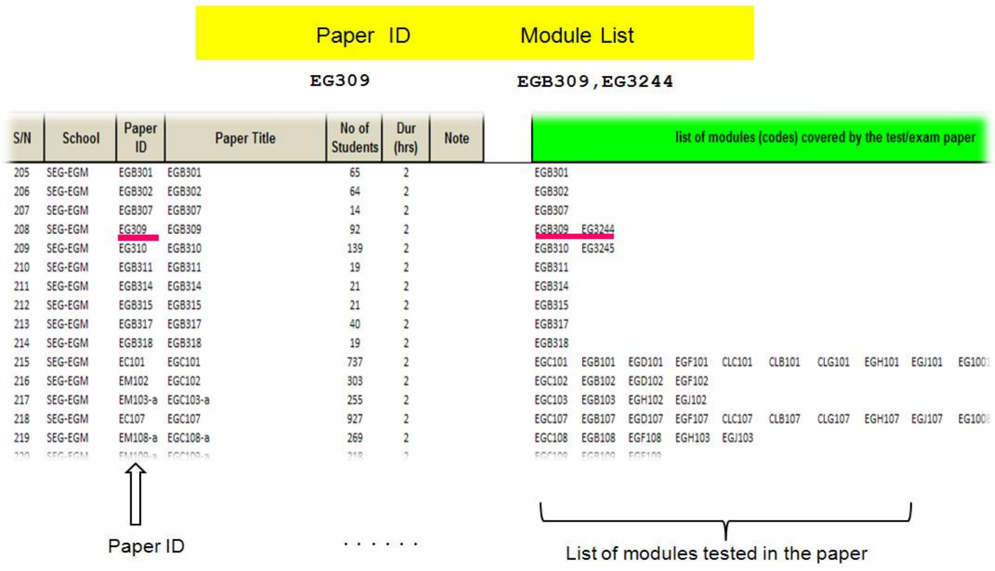
The planning period must be specified before the scheduling. Typical time slot specification is shown in Fig. 6. Note that in the exam scheduling, time is represented in form of integer numbers (namely slots). The numerical numbers are mapped back into real time for reporting purpose after schedule is complete.

Time Slots					(for Ref info only)
Day	Date	Week Day	Slot in Day	Time	Slot S/N (1 to N)
1	7-Feb-11	Mon	1	1000	1
			2	1500	2
2	8-Feb-11	Tue	1	1000	3
			2	1500	4
3	9-Feb-11	Wed	1	1000	5
			2	1500	6
4	10-Feb-11	Thu	1	1000	7
			2	1500	8
5	11-Feb-11	Fri	1	1000	9
			2	1500	10
6	14-Feb-11	Mon	1	0900	11
			2	1230	12
			3	1600	13
7	15-Feb-11	Tue	1	0900	14
			2	1230	15
			3	1600	16
8	16-Feb-11	Wed	1	0900	17
			2	1230	18
			3	1600	19
9	17-Feb-11	Thu	1	0900	20

Fig. 6 Time Slot Specification

Test Paper Information

The format of the test paper information in the IIEESS 1.0 system is shown in Fig. 7, which includes school code, paper ID; paper title; duration; and list of modules covered in the exam paper. The No. of students will be auto-counted by the system according to the student registration data set. Note that paper IDs must be unique within the whole schedule.



S/N	School	Paper ID	Paper Title	No of Students	Dur (hrs)	Note	list of modules (codes) covered by the test/exam paper
205	SEG-EGM	EGB301	EGB301	65	2		EGB301
206	SEG-EGM	EGB302	EGB302	64	2		EGB302
207	SEG-EGM	EGB307	EGB307	14	2		EGB307
208	SEG-EGM	EG309	EGB309	92	2		EGB309 EG3244
209	SEG-EGM	EG310	EGB310	139	2		EGB310 EG3245
210	SEG-EGM	EGB311	EGB311	19	2		EGB311
211	SEG-EGM	EGB314	EGB314	21	2		EGB314
212	SEG-EGM	EGB315	EGB315	21	2		EGB315
213	SEG-EGM	EGB317	EGB317	40	2		EGB317
214	SEG-EGM	EGB318	EGB318	19	2		EGB318
215	SEG-EGM	EC101	EGC101	737	2		EGC101 EGB101 EGD101 EGF101 CLC101 CLB101 CLG101 EGH101 EGJ101 EG100…
216	SEG-EGM	EM102	EGC102	303	2		EGC102 EGB102 EGD102 EGF102
217	SEG-EGM	EM103-a	EGC103-a	255	2		EGC103 EGB103 EGH102 EGJ102
218	SEG-EGM	EC107	EGC107	927	2		EGC107 EGB107 EGD107 EGF107 CLC107 CLB107 CLG107 EGH107 EGJ107 EG100…
219	SEG-EGM	EM108-a	EGC108-a	269	2		EGC108 EGB108 EGF108 EGH103 EGJ103
220	SEG-EGM	EM109-a	EGC109-a	218	2		EGC109 EGB109 EGF109

Fig. 7 Test Paper Information

5 Test Paper Conflictive Analyzer

After the student registration data and test paper information are loaded into the IIEESS 1.0 system, the system will perform test paper conflictive analysis which yields the mutually exclusive paper lists for every test paper as shown in Fig. 8.

For example, the paper titled "Materials Technology" with paper ID "EGC105", has a mutually exclusive paper lists as follows.

$$MEL_{EGC105} = EGB205{:}1, EGF212{:}1$$

This states that the paper EGC105 is mutually exclusive with EGB205 and EGF212. There is one common student who takes both EGC105 and EGB205; another student taking both EGC105 and EGF212. The system will not schedule the papers EGC105, EGB205 and EGF212 at same time to avoid the student conflicts. The mutually exclusive paper lists are also used when the system optimizes the searched solutions by minimizing the total number of common students involved in back-to-back (B2B) and multiple-papers-a-day (MPD) conflicts.

S/N	School	Paper ID	Paper Title	Stds	Dur	Nmzc	List of mutually exclusive paper lsit and common students				
1	SEG-EGM	EGC105	Materials Technology	302	1	2	EGB205:1	EGF212:1			
2	SEG-EGM	EGC111	Computer Programming	355	1.5	10	EGC205:1	EGC211:1	EGC204:2	EGB205:1	EGC210:1
3	SEG-EGM	EGC205	Manufacturing Information System	162	1.5	5	EGC111:1	EGC204:89	EGC305:2	EGC307:1	EGS213:6
4	SEG-EGM	EGC211	Computer-Aided Manufacturing/Enginee	121	1.5	5	EGC111:1	EGC210:92	EGC307:1	EGF206:23	EGS213:48
5	SEG-EGM	EGB204	Micro-controller Applications	144	1.5	6	EGB205:95	EGH204:37	EGB210:1	EGB211:1	EGB306:2
6	SEG-EGM	EGC204	Metrology & Quality Control	117	1.5	6	EGC111:2	EGC205:89	EGC210:1	EGC305:1	EGC307:1
7	SEG-EGM	EGB205	Quality Assurance	105	1.5	6	EGC105:1	EGC111:1	EGB204:95	EGB210:1	EGB306:1
8	SEG-EGM	EGF210	Metrology & Quality Control	26	1.5	2	EGF212:26	EGS213:13			
9	SEG-EGM	EGC210	Robotics System & Peripherals	102	1.5	4	EGC111:1	EGC211:92	EGC204:1	EGS213:47	
10	SEG-EGM	EGC305	Manufacturing Systems & Simulation	42	1.5	3	EGC205:2	EGC204:1	EGC307:31		
11	SEG-EGM	EGC307	Computer-Aided Design & Analysis	33	1.5	4	EGC205:1	EGC211:1	EGC204:1	EGC305:31	
12	SEG-EGM	EGD301	Shopfloor Monitoring & Control	82	1.5	5	EGC321:7	EGD211:3	EGD212:1	EGD308:26	EGF212:1
13	SEG-EGM	EGC312	Advanced Metrology & Quality Assuranc	22	1.5	0					
14	SEG-EGM	EGC321	Advanced Machining Technology	26	1.5	1	EGD301:7				
15	SEG-EGM	EGD211	Quality Process Control & Management	66	1.5	3	EGD301:3	EGD212:63	EGS213:26		

Fig. 8 List of mutually exclusive paper lists and conflictive coefficients

6 Exam Timetable Scheduler

The main GUI of the IIEESS v1.0 scheduler is shown in Fig. 9. The main features of the exam scheduler module are listed as follows.

- Load exam scheduling data and constraint information
- Auto-generate exam timetable solution
- Clash-checking and optimization support for manual operations
- Save searching results

Fig. 10 shows the drag-&-drop features provided for manual operations. Note that as the clash checking and constraint evaluation speed of the IIEESS 1.0, the system is generally many thousand times faster than the existing direct-clash0checking systems, it can provide efficient and effective support for manual operations. Typically the system can confirm a manual alteration made to the schedule in few micro-seconds, whereas old systems can take several minutes (as long as 30 minutes) to confirm a change made to the schedule.

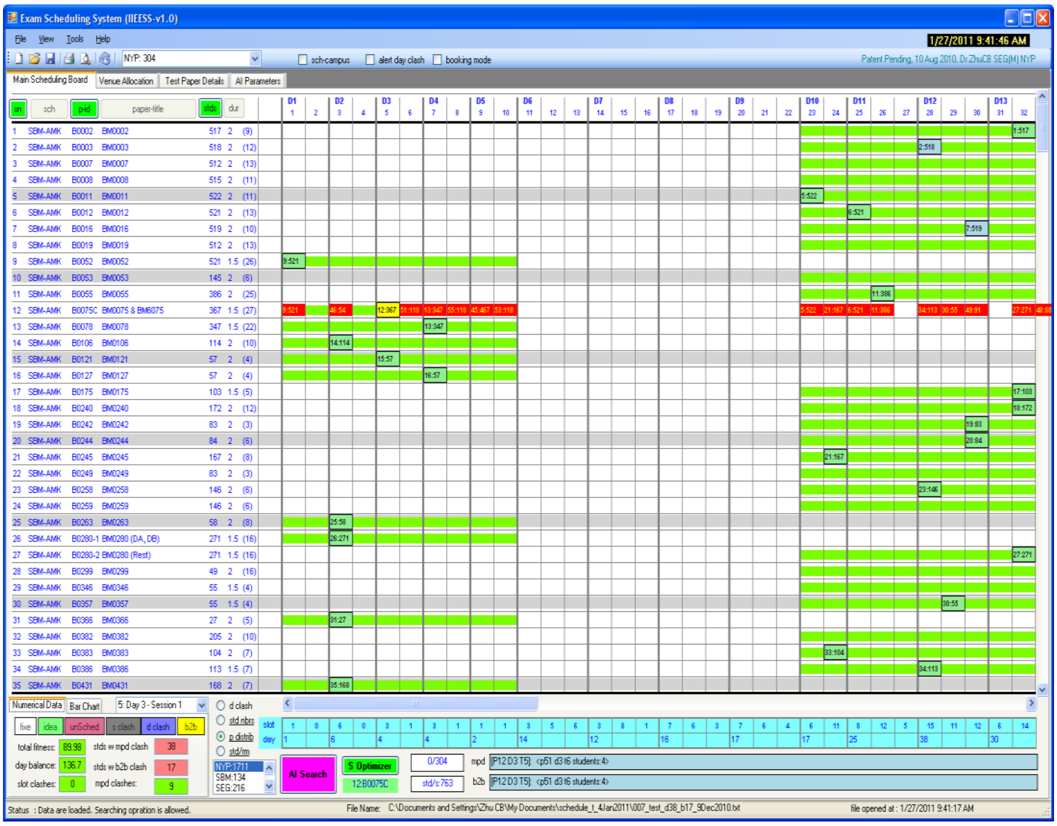


Fig. 9 IIEESS v1.0 Main –Scheduler

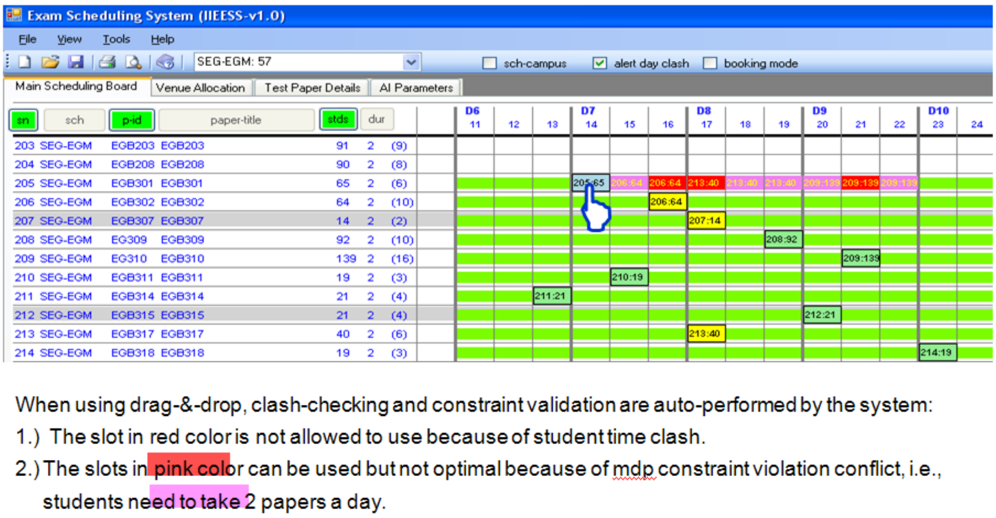


Fig. 10 Drag-and-drop features for manual operations on a schedule

7 Venue Resource Allocation Module

The un-facilitated exam activities which are scheduled using AI scheduler are then facilitated with venue resources by the venue allocation module (see Fig. 11). The module provides an auto searching mode and a manual drag-&-drop feature for manual venue allocation.

The fully facilitated exam schedule is saved a file as a final solution that is ready for reporting or exporting to the legacy database system.

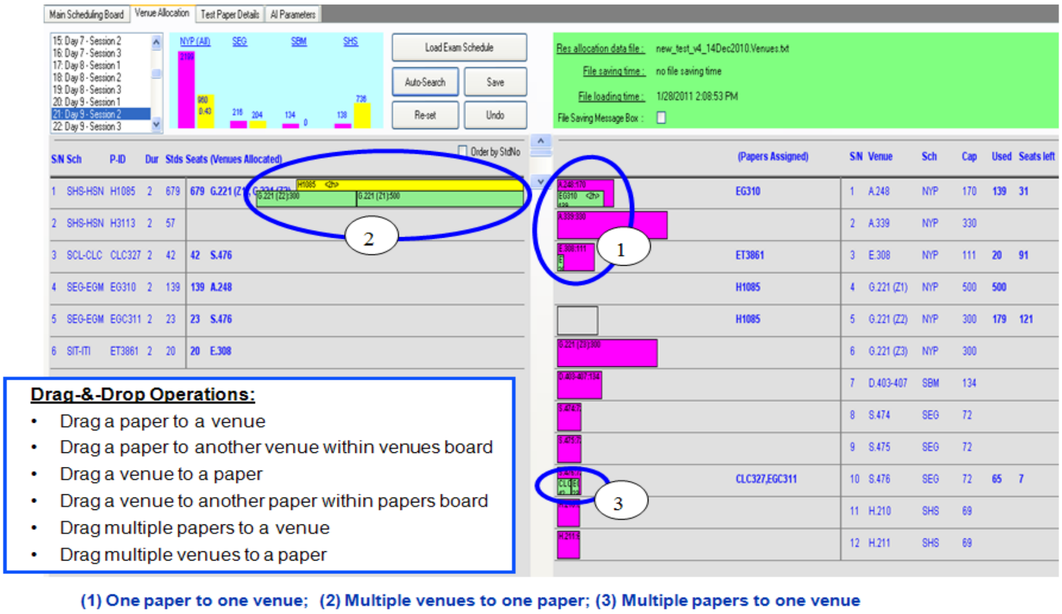


Fig. 11 Drag-and-drop features for resource allocation

8 Exam Timetable Reporting

There are two ways to generate the exam schedule reports. Firstly, the IIEESS 1.0 system can upload the schedule solution to a legacy database if any, so that the existing reporting tools can be used for the students and staff to access the exam schedule, as can be seen from Fig. 8. Secondly, the system can generate the required exam timetables directly using solutions created by the system, as can be seen from Fig. 12 and 13.

9 Main Features

1.) Most suitable for Large exam scheduling problems, with complex cross school/department registrations (no of candidates can be as high as many thousands), solutions are accurate and robust.
2.) User-friendly registration data entry and system parameter setting; ease constraint and scheduling requirement specification.
3.) Fast solution searching (runtime is within few minutes for a sizable exam scheduling problem).
4.) User-friendly drag-&-drop features, conflict checking is done in few micro-seconds.
5.) Advanced reporting and integration tools for schedule output and statistics.

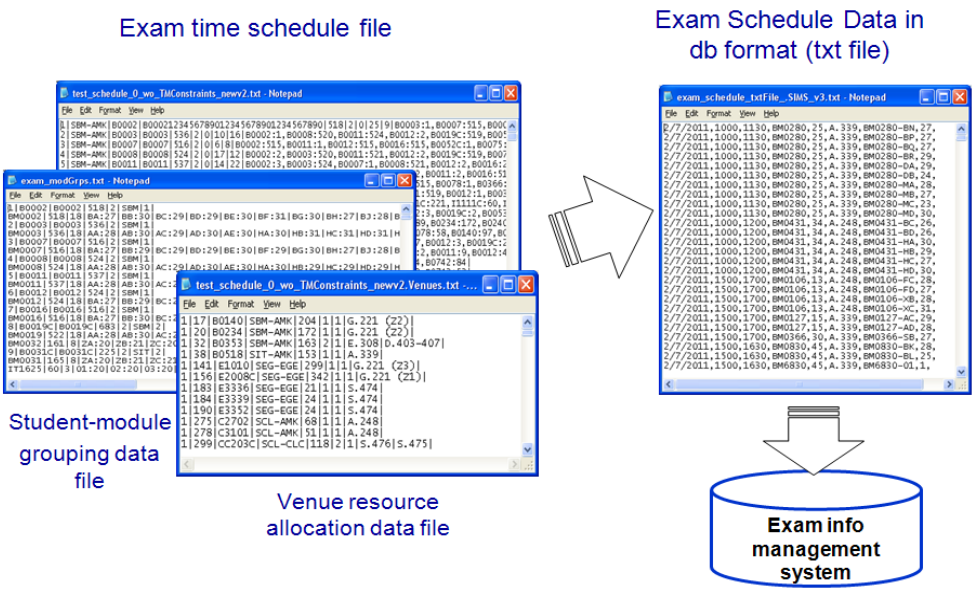


Fig. 12 Export the exam schedule to database

Exam Timetable (Summary)

Date	Time	Paper	Module	Groups	StdNo	Venue	Senior Inv
6/27/2011 Mon	0830-1000	Engineering Mathematics 1A/B	EGC101	C1-C4	91	LTR1	
			EGC101/EGB101	C5,B1-B2	68	LTQ1	
			EGB101	B3-B5	66	LTQ2	
			EGD101	D1-D3	56	LTQ3	
			EGF101/EGH101	F1,H1-H2	65	LTQ4	
			EGJ101/CLC101	J1,C1-C2	70	LTQ7	
			CLC101/CLB101/CLG101	C3,B1,G1	72	LTQ8	
		Aerospace Material & NDT Technology	EGC322/EGF301	C1,F1-F2	42	LTQ9	
	1100-1230	Engineering Mathematics 2C	EGC207	C1-C3	65	LTQ7	
			EGC207/EGB207	C4-C5,B1	62	LTQ8	
			EGB207	B2-B4	62	LTQ9	
			EGB207/EGD207	B5,D1-D2	63	LTQ11	
			EGD207/EGF207/EGJ207	D3,F1,J1	70	LTQ12	
		Industial Safety & Loss Prevention	CLC301	C1-C3	63	LTQ1	
			CLB301/CLG301	B1,G1	44	LTQ2	
		Product Innovation & Rapid Prototyping	EGC310/EGD306	C1,D1-D2	69	LTQ3	
		Automated Machine Design	EGC302	C1-C2	35	LTQ4	

Fig. 13 Exam Timetable Schedule Report

10 Hardware/Software Requirements

Hardware: IBM PCs, laptops or equivalents; monitor resolution 1680x1060.
Software: MS Windows XP or above, MS .NET framework 2.0, or above, MS Office 2005 or above.

11 Conclusions

The IIEESS v1.0 system is *highly efficient*; it is much faster than traditional direct-clash-checking systems in terms of solution searching and constraint validation. The system particularly performs well in scheduling large number of exam activities with large number of candidates who registered with multiple modules (papers) cross schools or departments. The runtime for a sizable exam scheduling problem is extremely fast, e.g., in few minutes.

The system is truly *interactive*. It provides user-friendly drag-&-drop features for the planners to book a time slot for an exam before scheduling or to amend the schedule after scheduling. When booking a slot for an exam or amend the schedule auto-generated using the system, full supports will be provided by the IIEESS v1.0 system, such as clash-checking and optimizing solution searching and satisfying constraints imposed. The changes made can be confirmed in few micro-seconds. The new system is *transparent* and *robust*. The IIEESS 1.0 system is always able to generate a solution. If no complete solution exists, the system generates an in-complete solution and indicates the un-scheduled papers and displays reasons why they cannot be scheduled, such as like conflict constraints being imposed.

Although the IIEESS 1.0 system is designed for exam scheduling, the new method and patent technology can be applied into a wide range of other applications, such as transportation planning, sports activity scheduling, vehicle routing and man power scheduling in various production and service industries where event conflictive features and relations must be analyzed before solution searching.

References

Nelishia Pillay and Wolfgang Banzhaf, (2007). "A Genetic Programming Approach to the Generation of Hyper-Heuristics for the Incapacitated Examination Timetabling problem", in Progress in Artificial Intelligence, 13th Portuguese Conference on Artificial Intelligence, EPIA 2007, Springer, ISBN: 978-3-540-77000-8, pp.223-234,

Asmuni, H., Burke. E.K., Garibaldi, J.M. (2005). "Fuzzy Multiple Ordering Criteria for Examination Timetabling", vol.3616, pp.147-160. Springer, Heidelberg.

M. Dorigo, and V. Maniezzo, and A. Coloni. Ant System, (1996). "Optimization by a Colony of Cooperating Agents". IEEE Trans. Sys., Man, Cybernetics 26(1996) 1, p29-4.

Goldberg, David E. (1989), "Genetic Algorithms in Search Optimization and Machine Learning". Addison Wesley. pp. 41. ISBN 0201157675.

Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948, ICNN.1995.488968.

Kennedy, J.; Eberhart, R.C. (2001). Swarm Intelligence. Morgan Kaufmann. ISBN 1-55860-595-9.

Shi, Y.; Eberhart, R.C. (1998). "Parameter selection in particle swarm optimization". Proceedings of Evolutionary Programming VII (EP98). pp. 591–600.

Eberhart, R.C.; Shi, Y. (2000). "Comparing inertia weights and constriction factors in particle swarm optimization". Proceedings of the Congress on Evolutionary Computation. 1, pp. 84–88.

Shu-Chuan Chu, Yi-Tin Chen, Jiun-Huei Ho, (2006). Timetable Scheduling Using Particle Swarm Optimization, Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06), 0-7695-2616-0/06, © 2006 IEEE.

UniTime LLC, Examination Timetabling Problem Description, © 2008 - 2011 UniTime LLC, http:// www.unitime.org/exam_description.php.

Examination Timetabling Problem Description, http://www.unitime.org/exam_description.php

Zhu Chunbao, (2008). US patent No. 7,447,669, “Method and System for Timetabling Using Pheromone and Hybrid Heuristics Based Cooperating Agents”, November 4, 2008.

Zhu Chunbao, (2010). PCT Patent Application Number PCT/SG2010/000388, “Method and System for Examination Timetable Scheduling Using Test Paper Conflictive Analysis and Swarm Intelligence”, November 1, 2010.