Graph-based Conflict Rectification using Stroke Gesture Approach in Timetabling System (CORECTS)

Yong Tien Fui[1], Choo Wou Onn[1], Chue Wen Yeen[2], Kok Hui Meian[3]

[1]Department of Knowledge Science
Faculty of Information and Communication Technology
[2]Department of Finance
Faculty of Business and Finance
[3]Department of Advertising
Faculty of Arts and Social Science

Universiti Tunku Abdul Rahman (UTAR),
Jalan Universiti, Bandar Barat, 31900 Kampar,
Perak, MALAYSIA
yongtf@utar.edu.my, choowo@utar.edu.my, chuewy@utar.edu.my, kokhm@utar.edu.my

ABSTRACT

In this paper, the proposed system, CORECTS, is specially designed and developed to check if conflicts exist within planned timetable. Although this problem seems simple at first glance, it is actually quite complex to resolve it if the conflicts are not presented clearly to the schedule planner. Thus, a special graphical user interface is designed to make the entire system being more efficient to deal with conflicts when detected. CORECTS will apply a hybrid method using annealing graph algorithm and the proposed STAR-quadrants to display the conflict detection of the entire timetable. Simple stroke gesture is used to expedite the conflict resolution process. CORECTS is aimed to ease the work of the user, not just by making timetable planning more interesting, but also making some tedious configuration works faster and thus improves efficiency.

Keywords
Timetabling System, Scheduling, Conflict Detection, Graph Annealing, Stroke Gesture

1.0 INTRODUCTION

Creating a course timetabling system has been an ongoing challenge even for today. This is a hard problem because of its size and the complexity of constraints needed to satisfy the demands of students and instructors. One of the greatest challenges, among other, is that no completely perfect solution can be achieved through the current fully automated timetabling system. It may be due to limitation of the current techniques to deal with very complex constraints required by many continuously expanding universities. The research direction in the field of timetabling system has been along the pathway to find better approaches to build an intelligent system that are fully automated and with hope to replace human works in the area. Rather than approaching the solution in that way, it is vital to look for alternative solution that can best help timetable planners to perform their work faster and more productively.

CORECTS is specially designed and developed to check if conflicts exist within the planned timetable. For example, the same academic usually cannot teach two or more subjects with overlapping time periods, and two courses with overlapping time periods cannot use the same classroom. Although this problem seems simple at first glance, it is actually quite complex to resolve it if the conflicts are not presented clearly to the schedule planner. Table 1 and Table 2 show timetables for two different lecturers. Although Combs et. al. (2005) has quoted that an academic course timetable should ideally be displayed as a grid (usually with days of the week on one axis and time periods on the other) due to that this format is familiar and facilitates visual checking for scheduling conflicts. This format, however, does not make conflict handling easy to be implemented as the relationships of the conflicts cannot be visually presented on one screen to facilitate the rectification process. Thus, a special graphical user interface is designed to make the entire system being more efficient to deal with conflicts when detected.

Table 1: Timetable of a lecturer teaching two subjects, presented in grid form, with conflicting areas shaded in red.

Lecturer/Tutor Name: Albert Yong

	8am-9am	9am-10am	10am-11am	11am-12pm	12pm-1pm	1pm-2pm	2pm-3pm	3pm-4pm	4pm-5pm	5pm-6pm	6pm-7pm	7pm-8pm
Monday												
Tuesday												
Wednesday					UCCD2000 H112 (T1)	UCCD2000 H220 (L2)	UCCD3000 H109 (T1)					
Thursday	UCCD2000 H112 (T2)	UCCD2000 IDK8 (L1)								UCCD3000 H109 (L1)		
Friday				UCCD3000 H220 (L2)			UCCD2000 H220 (T3)			UCCD3000 H109 (T2)		

Table 2: Timetable of another lecturer with conflicting areas shaded in red.

Lecturer/Tutor Name: Jass Kok

	8am-9am	9am-10am	10am-11am	11am-12pm	12pm-1pm	1pm-2pm	2pm-3pm	3pm-4pm	4pm-5pm	5pm-6pm	6pm-7pm	7pm-8pm
Monday												
Tuesday												
Wednesday												
Thursday												
Friday				UCCD1000 IDK8 & H220 (T2) & (T1)		UCCD1000 H220 (L1)						

The CORECTS will apply simulated annealing graph algorithm to simulate the conflict detection of the entire timetable and be presented on a touch screen monitor where the problems can be resolved via simple stroke gestures made by the planner. The CORECTS is aimed to ease the work of the user, not just by making timetable planning more interesting, but also making some tedious configuration works faster and thus improves efficiency.

This paper first reviews the literature related to the university course timetabling system problems. Research objectives will be discussed in Section 3. Section 4 presents the contribution. The conclusion will appear in Section 5.

2.0 RESEARCH BACKGROUND

The proposed conflict detection of timetabling problems contains three parts. Firstly, the timetable has to be presented in a form that will make it easy for the schedule planners to detect conflicting areas. Section 2.1 highlights the problem with traditional approach in timetabling. Next, Section 2.2 presents graph annealing techniques used to draw graphs with no overlapping nodes. Finally, Section 2.3 presents stroke gesture approaches to improve interaction between users and computer systems.

2.1 Timetabling System

Ho Sheau Fen et. al. (2009) argued that timetable planning is a complex problem and cannot be dealt with lightly. They explained the complicated associations between three common parameters, such as time periods, subjects and classrooms, make it difficult to obtain feasible solution. Thus, finding feasible solution for timetable planning is a still a challenging problem. In this paper, our focus is centered around representing timetable information visually to facilitate conflict detection. We assume the timetable is already produced and ready for conflict checking.

Yan Yang (2006) mentioned that there are fundamentally two open issues in the course timetabling problem; the application-specific issues and the dynamic-constraint issue. The former issue is one in which a solution to one course timetabling problem may not be suitable for another one, while the latter issue addresses the course timetabling problem becomes much harder when the constraints change dynamically. Although they proposed a general solution model using agent technology, it is certainly more viable solution for allowing human schedule planner to solve this kind of problem. To do this, there is a need to represent information in a way that the user finds it easy to have a holistic view of the overall situation or problem.

Representing information using grids and tables have been a traditional approach in displaying areas of conflict in timetabling checking, whether in manual or computerize form. Even if an academic timetable is constructed manually or by an intelligent system, it is important to check it for conflicts or other problems autonomously. Combs et al. (2005) suggested that it would likely be useful to model a schedule as a graph for conflict checking purpose; however, they proposed a model which uses grids to check for conflicts.

2.2 Graph Annealing

Davidson and Harels (1996) proposed a simulated annealing graph algorithm to apply to the problem of drawing graph nicely (as in Figure 1), which can be used to produce graph showing conflicting areas in the course timetable.

Applications that use graphs often have nodes with labels. Xiaodi Huang et. al. (2007) proposed an approach to make the label of each node readable, making the drawings of the graphs appeared to have no overlapping nodes. He stated that it is crucial in a dynamic environment of graph visualization. He proposed an approach that employs Force-Transfer algorithm that removes node overlaps.

Satu Elisa Schaeffer (2007) presented several algorithms for producing clusters for an input graph and share some thoughts on the application areas of graph clustering algorithms.

Yi-Yi Lee et. al. (2006) took a step further by presenting a simulated annealing graph drawing algorithm that can preserve what they called the mental map. The mental map preserves the contour and relative positions of the redrawn graph when modification is made to the original graph.

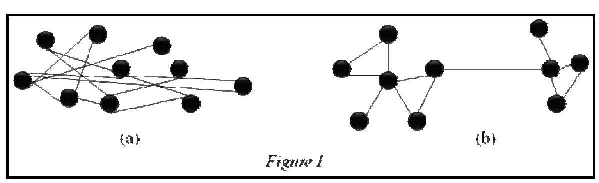

Figure 1

Figure 1(a) is a typical input graph and Figure 1(b) is the output produced by the simulated graph annealing technique.

2.3 Stroke Gesture

The detected conflicts appeared visually as a graph, can then be corrected easily and quickly using simple trained strokes gestures on a touch-screen monitor, as shown in Figure 2. Appert and Zhai (2009) proposed to using strokes as Command Shortcuts. Bau et. al. (2008) suggested that gesture-based interfaces allow users to interact with the objects of interest on the screen directly from the current cursor position, providing users with a direct and efficient form of interaction compared to buttons and pull-down menus, which force the user to move the mouse or pen to the command's location on the screen. As the process is repeated many times during the course of action to rectify the conflicts, a lot of time is obviously wasted unproductively. Hinckley (2006) proposed an approach which can offer a single gesture command that acts on a wide variety of selections. This ability to select objects using multiple strokes and thus allows additional functionality, including more precise control over multiple object selections, easier editing of selections, and nesting of multiple operations within a selection. It can streamline some operations by removing the necessity to explicitly group and ungroup objects, potentially improving the cognitive and physical workflow of such operations.

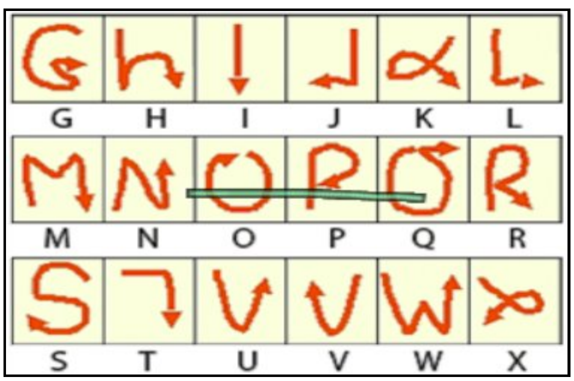

Figure 2: Sample stroke gestures used on PDAs and touch screen monitors (Source: http://www.yorku.ca).

3.0 RESEARCH OBJECTIVES

The objectives of the CORECTS are to develop a prototype for the best conflict checking model by using graphic-based and stroke gesture approach. In the mean time, the usefulness of the prototype as a supporting tool to check for conflict in the existing timetabling system will be evaluated.

We are now proposing a modified graph annealing technique to show conflicting areas in timetabling problem. We are not focusing on creating a new timetabling system; rather, we are proposing a supporting tool to verify whether conflicts arise within planned timetable. It can work on timetable information produced manually or using automated timetabling system. We assume a full set of timetable exist before our proposed system can be used for the conflict detection.

4.0 CONTRIBUTIONS

We propose a modified graph annealing technique to show conflicting areas in timetabling problem, as shown in Figure 3. We assume a full set of timetable exists before our proposed system can be used for conflict detection.

Based on the details provided to by the timetable planner, the timetabling file is read as input data into CORECTS. The proposed system starts analyzing the available data to detect for conflicts, such as same academic staff teaching at two different venues concurrently, or two entire different subjects are held in the same room, to name a few. Another example, three lecturers have been assigned to three different subjects. However, all the lecturers are assigned to conduct classes in Room H220 at the same time and same. It is obvious that there is conflict in room assignment. This kind of problem can be difficult to detect if not presented in graph form. In this case, the Room quadrant for the three conflicting nodes will be colored red to indicate problem in those areas, as shown in Figure 5.

When there is at least one area conflicting with the same of other nodes, all conflicting nodes will be colored in red color. When user interested to view the conflicting area in greater detail, the user can zoom in to the conflicting area.

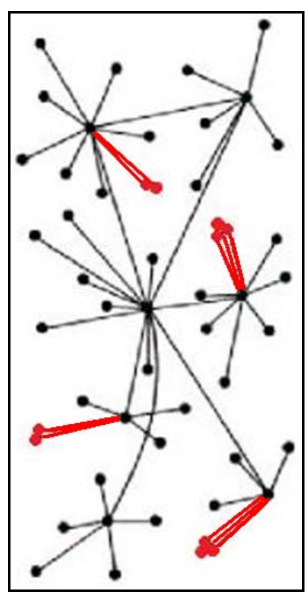

Figure 3: Overview of timetable and conflicting areas using graph representation.

In the detail view, the user will see each node having four distinct quadrants, known as STAR-quadrants (Figure 4), where:

- **S** represents Subject Code. Within the Subject quadrant, it has a sub-area to record lecture/tutorial groups for a given course;
- **T** represents Time slots (from time slot and to time slot) of a class. The acceptable time slot codes can be found in Table 3. Within the Time quadrant, it has a sub-area to record the duration of the class;
- **A** represents Academic staff name. Within the Academic quadrant, it has a sub-area to record any constraints given by the lecturers/tutors. For example, some travelling lecturers prefer not to teach on every Friday of the week, while some other have specific requirement that they wish the timetabling planner to take note. All the constraints are entered in this sub-area; and
- **R** represents Room number. Within the Room quadrant, it has a sub-area to record the room capacity and available places for student to register for a given lecture/tutorial class.

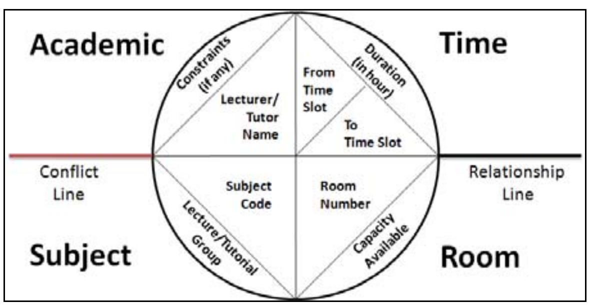

Figure 4: A typical STAR-quadrant node used in the proposed method, which contains common timetabling information plus additional details such as capacity available for student registration, teaching staff's constraints, and conflicting areas line. The four quadrants are Subject (bottom left), Time (top right), Academic (top left) and Room (bottom right).

Table 3: Coded time slots for entry into the Time quadrant on a node.

	8am-9am	9am-10am	10am-11am	11am-12pm	12pm-1pm	1pm-2pm	2pm-3pm	3pm-4pm	4pm-5pm	5pm-6pm	6pm-7pm	7pm-8pm
Monday	1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9	1-10	1-11	1-12
Tuesday	2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8	2-9	2-10	2-11	2-12
Wednesday	3-1	3-2	3-3	3-4	3-5	3-6	3-7	3-8	3-9	3-10	3-11	3-12
Thursday	4-1	4-2	4-3	4-4	4-5	4-6	4-7	4-8	4-9	4-10	4-11	4-12
Friday	5-1	5-2	5-3	5-4	5-5	5-6	5-7	5-8	5-9	5-10	5-11	5-12

In Figure 6, the entire graph is drawn based upon graph annealing technique to improve the clarity of the problems being presented. It appears that the graph is neater and nicer compared to the one presented in Figure 5, where all the nodes seem intertwined with one another. We used a modified graph annealing technique, coupled with STAR quadrants sticking above each node to represent the holistic view of the whole timetable. By doing so, each node carries greater amount of information to help schedule planners perform their work more efficiently and effectively, thus making timetabling work less distasteful but more enjoyable than before. Note that the graph annealing technique used is able to group related subject together, despite the possibility of having different time slots, venues, and/or academic staff teaching it. The grouping is represented using dotted blue line.

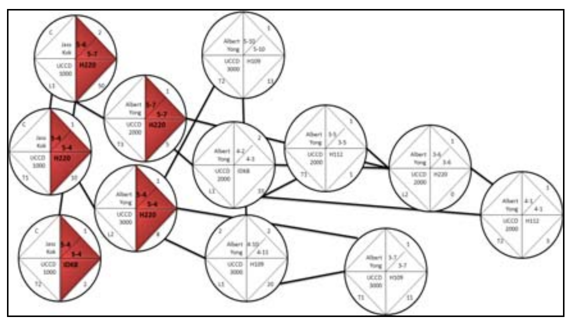

Figure 5: Detail view of the timetable's conflicting nodes detected. Time and Room quadrants are the causes of the conflicts in all cases and appeared red in color to indicate the type of conflict arises. The graph is produced from Table 1 and Table 2.

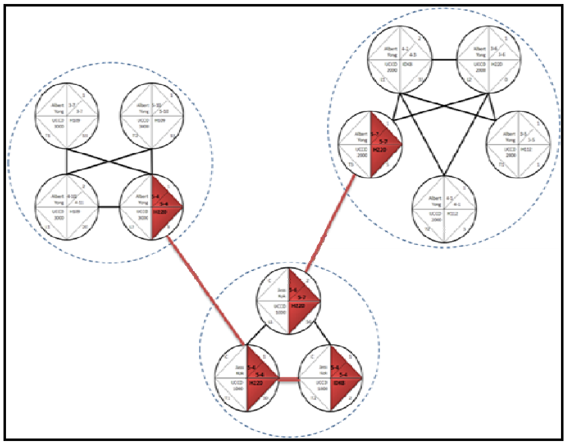

Figure 6: Graph representation of the same conflicting areas as in Figure 5, but with graph annealing technique applied to make the drawing appears nicer and neat.

The user can zoom in (or out) to magnify (or reduce the size) of the graph, as shown in Figure 7. When the conflicting areas are focused, the user can use simple stroke gesture on the monitor screen to perform some predefined actions on the selected node or quadrant of the node. For example, the user can tap twice on the Room quadrant of the left-most node in Figure 7 to change the location. This can be done by selecting a new location from a list that appears in a pop-up window. All the pop-up lists, such as lecturers, subjects, time slots, room numbers, constraints, lecture/tutorial groups, and capacity available are stored and retrieved from a relational database management system.

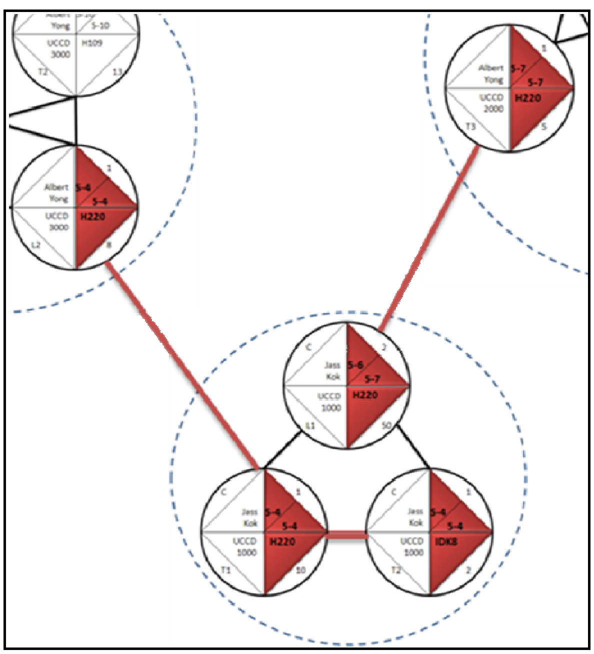

Figure 7: Zoomed-in view of the areas of conflicts, where any node or quadrant can easily be selected for an action to be performed on it using simple stroke gesture.

Besides tapping on the objects of interest, the user can also use other stroke gestures to achieve various other operations, as shown in Figure 8.

6.0 CONCLUSION

CORECTS is specially designed and developed to check if conflicts exist within planned timetables. CORECTS is aimed to ease the work of the user, not just by making timetable planning more interesting, but also making some tedious configuration works faster and thus improves overall efficiency. A modified simulated annealing graph algorithm makes it possible for timetabling information be presented in a much clearer format than with grids or tables. Conflict detection can be done fairly easily when conflicting nodes are grouped together to form clusters of vertex. Stroke gesture is the applied to help better interaction between the users and the proposed system to resolve the conflicts quickly and conveniently. Since future operating systems will be supporting touch screen, we found that stroke-gesture will be a popular technique to resolve not just timetabling conflicts, but also many other applications.

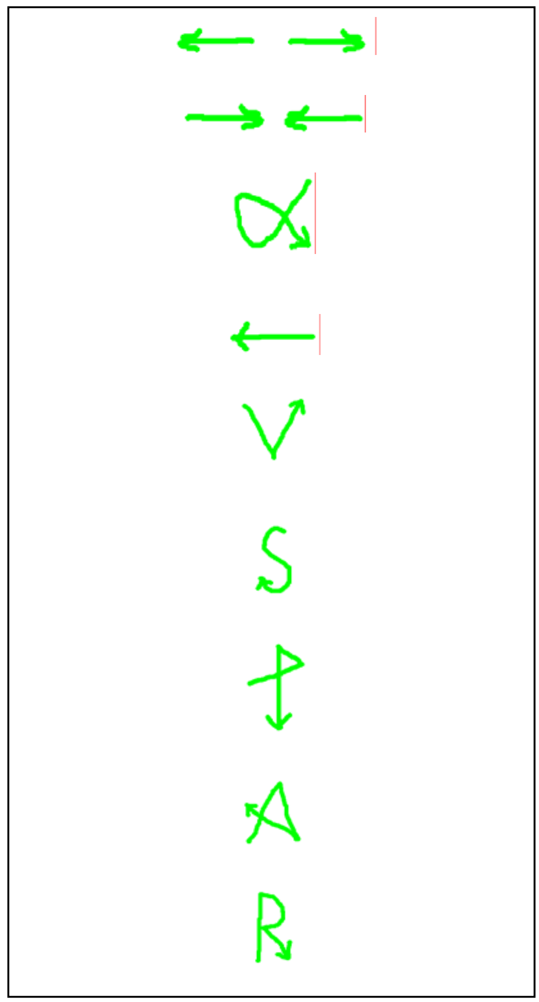

Figure 8: Simple stroke gesture to command the proposed system to perform an action.

REFERENCES

Appert, C., & Zhai, S. (2009). Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support. *In ACM CHI 2009, April 4 - 9, Boston, Massachusetts, USA.*

Bau, O., Mackay, & W. E., OctoPocus. (2008*)*. A Dynamic Guide for Learning Gesture-Based Command Sets. *In ACM UIST'08, October 19-22, Monterey, California, USA.*

Combs, W., Hawkins, R., Pore, T., Schechet, A., Wahls, T., & Ziantz, L. (2005). The Course Scheduling Problem as a Source of Student Projects. *In ACM SIGCSE'05, February 23–27, St. Louis, Missouri, USA.*

Davidson, R., & Harel, D. (1996). *Drawing Graphs Nicely Using Simulated Annealing*. *In ACM Transactions on Graphics, 15*(4), 301–331.

Fen, H. S., Safaai, D., Hashim, M., & Siti Zaiton. (2009). University Course Timetable Planning Using Hybrid Particle Swarm Optimization. *In ACM/SIGEVO Summit on Genetic and Evolutionary Computation.*

Hinckley, K., Guimbretiere, F., Agrawala, M., Apitz, G., & Chen, N. (2006). Phrasing Techniques for Multi-Stroke Selection Gestures. *In ACM Graphics Interface.*

Huang, X., Lai, W., Sajeev, A.S.M., & Gao. J., (2007). A New Algorithm for Removing Node Overlapping in Graph Visualization. *In Elsevier, Information Sciences, 177*, 2821–2844.

Satu Elisa Schaeffer. (2007). Graph Clustering. *In Elsevier, Computer Science Review, 1*, 27-64.

Yang, Y., Paranjape, R., & Benedicenti, L. (2006). An Agent Based General Solution Model for the Course Timetabling Problem. *International Conference on Autonomous Agents.*

Yi-Yi Lee, Chun-Cheng Lin, Hsu-Chun Yen. (2006). Mental Map Preserving Graph Drawing Using Simulated Annealing. *In Asia Pacific Symposium on Information Visualization 2006 (APVIS 2006), 60.*