

# USING GRID PUZZLE TO SOLVE CONSTRAINT-BASED SCHEDULING PROBLEM

Noppon Choosri

SMART Research Centre, College of Arts Media and Technology,  
Chiang Mai University, Chiang Mai, Thailand  
noppon.c@cmu.ac.th

## **ABSTRACT**

*Constraint programming (CP) is one of the most effective techniques for solving practical operational problems. The outstanding feature of the method is a set of constraints affecting a solution of a problem can be imposed without a need to explicitly defining a linear relation among variables, i.e. an equation. Nevertheless, the challenge of paramount importance in using this technique is how to present the operational problem in a solvable Constraint Satisfaction Problem (CSP) model. The problem modelling is problem independent and could be an exhaustive task at the beginning stage of problem solving, particularly when the problem is a real-world practical problem. This paper investigates the application of a simple grid puzzle game when a player attempts to solve a practical scheduling problem. The examination scheduling is presented as an operational game. The game's rules are set up based on the operational practice. CP is then applied to solve the defined puzzle and the results show the success of the proposed method. The benefit of using a grid puzzle as the model is that the method can amplify the simplicity of CP in solving practical problems.*

## **KEYWORDS**

*Constraint Programming; Constraint Satisfaction Problem; Examination scheduling; Grid puzzle*

## **1. INTRODUCTION**

Constraint Programming (CP) is a programming paradigm used for modelling and solving problems with a discrete set of solutions [1]. The idea of the CP is to solve problems by stating a set of constraints (i.e. conditions, properties or requirements) of the problems and finding a solution satisfying all the constraints using a constraint solver [2, 3]. The main advantage of the CP approach is the declarative ability of the constraints which makes it suitable for solving complicated real-life problems. In order to solve the problem using CP, a model is required and it is typical to define the problem as Constraint Satisfaction Problem (CSP). CSP is defined by a sequence of variables. A finite sequence of variables  $Y := y_1, \dots, y_k$  where  $k > 0$ , with respective domains  $D_1, \dots, D_k$ . A finite set  $C$  of constraints are used to limit the domain for each variable [4]. There is another problem called Constraint Satisfaction Optimisation Problem (CSOP) which can be seen as an 'upgrade' of CSP in the sense that solutions are not only feasible but also achieve optimality of an integrated cost function [5]. Formalism of CSP is defined in [6]. Typically, to solve practical operational problems using CP, ones are only required to model the

Natarajan Meghanathan et al. (Eds) : ICAIT, ICDIPV, ITCSE, NC - 2014  
pp. 09–18, 2014. © CS & IT-CSCP 2014

DOI : 10.5121/csit.2014.4702

problems and using CP solvers to solve the problems. There are several available CP solvers for both CSP and CSOP including: Choco, Ilog, ECLiPSe®, Gecode, Comet, CHIP, and Jsolve. Problem modelling is one of the key steps of using CP to solve problems successfully. This paper will focus on a grid puzzle-game as inspiration to model and solve the problem. The rest of the paper is organised as follows; Section 2 discusses the current CP applications, Section 3 provides a background of typical grid puzzle game, Section 4 demonstrates the using of grid puzzle to model a scheduling problem, Section 5 discusses the CP implementation, Section 6 discussed the result of the paper and, Section 7 is the conclusion.

## 2. CP APPLICATIONS

CP has been applied to solve several applications successfully. In healthcare, CP is used to assign shifts to medical staffs. Several rules can be imposed to solve the problem and create the realistic schedule including; assignments meet the demand for every shift, staff availability status, and the fairness of the generated schedule for every assigned staff [7]. Further requirements to schedule working time for medical residents are addressed in [8]. The requirements that make this scheduling different from the typical medical staff come from the fact that a resident is not only the medical staff, he or she is also a student in training i.e. the schedule have to provide a good balance between education and medical service activities. CP is also used for scheduling facilities in healthcare such as an operation theatre[9]. At airports, [10] investigates the use of CP to schedule aircraft departure to avoid traffic congestion, while [11] focuses the study on generating a contingency plan to handle unexpected failures affecting a regular traffic schedules. At academic institutes, manual timetabling can be a very time-consuming task, [12] presents CP based school timetabling to minimise idle hours between the daily teaching responsibilities of all teachers. [13] develops an examination timetabling to tackle important constrains such as schedule clashing, room capacity, and avoiding an allocation of two difficult subjects in consecutive time slot.

## 3. GRID PUZZLES

Grid Puzzles are board games contained within an  $N \times M$  lattice where players are usually required to locate symbols or number to meet the objective of the game. There have been several studies using CP to solve grid puzzle games. Akari, Kakuro, Nurikabe have been studied [14]. Akuro is a game that provides clues for a number of tokens, which the game called 'lights', for certain grid, players are asked to locate tokens such that all conditions are satisfied. Kakuro requires players to fill a numbers to grids to generate sums to meet vertical and horizontal clues. Another classical puzzle game problem that is usually mentioned in CP literature is the  $N$ -queen problem. In this problem, one is asked to place  $N$  queens on the  $N \times N$  chess board, where  $N \geq 3$ , so that they do not attack each other. Better known puzzle games are Crosswords and Sudoku, and MineSweeper. Crosswords are games in which one is required to fill pre-defined vocabulary into the  $N \times N$  grids in a way that none of the words are used more than once. Sudoku is usually played on  $9 \times 9$  grids with some grids having pre-defined values. The game's rule involves giving a value assignment so that all rows and column as well as sub-regions  $3 \times 3$  grid are pairwise different. Finally, Minesweeper is one of the most popular 'time-killer' computer games which has the objective to determine the 'mine' on a grid where the game might provide hints for a number of mines in the grids. The example of the Grid puzzle games are shown as Figure 1.

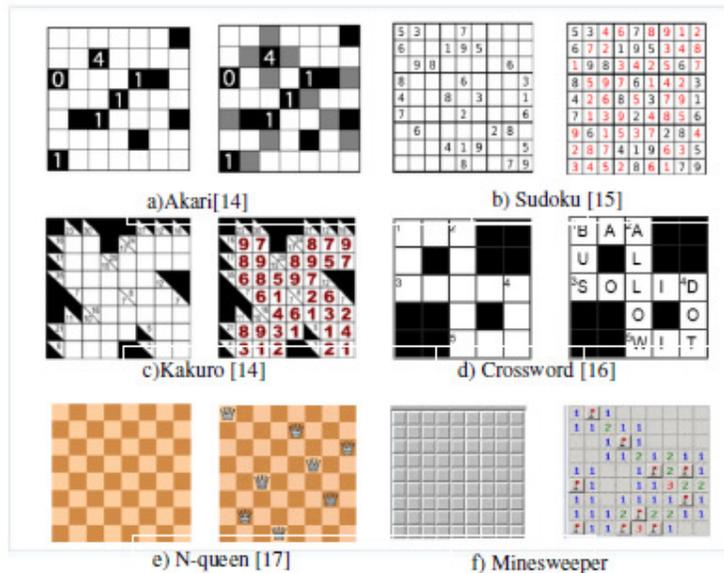


Figure 1. Typical grid puzzle games and their solutions [14-17]

#### 4. CP APPLICATIONS

The mechanism of tackling CSP using CP typically relies on the domain reduction process. To solve a problem, a set of constraints related to the problem needs to be identified and later on applied to a problem. Some of the constraints are associated with each other to formulate a constraint network. Each constraint applied to the model is usually associated with finite domain variables. Solving the problem is a process of reducing the domain of each variable until there are no conflicted domains remaining. So, constraint programmers will need to understand the variables, domain and constraints of the problem. Particularly they need to have a comprehensive understanding of the relationship among associated constraints and variables. This can be an exhaustive task when solving complicated practical problems. Figure 2 visualises an abstraction of a constraint network and variable network of CP as describe above.

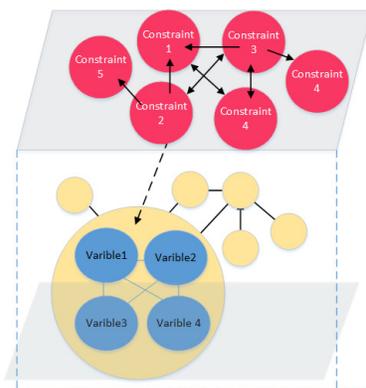


Figure 2. CP problem solving

Grid puzzles representations, i.e. using 2 Dimension (2D),  $N \times M$  , lattice to represent finite values/states of variables, which can be applied to model many practical problems. With that, the relationship between variables can be visualised. Rules of the games can be set up to

reflect businesses rules, and typical constraints can be applied to the model just as what shown in solving general puzzle games. This paper demonstrates the use of grid puzzles for solving an examination scheduling problem which is outlined as follows:

**Problem definition:** The problem is an examination scheduling problem. It is mainly concerned with assignment of subjects for exam into given time slot during examination period. The generated result shall be able to indicate the day of the week the exam is allocated together with the room assigned. The assumption of this problem is that this schedule is for a package registration system in which student in the same year will study the same subjects. The problem is concerned with practical constraints such as certain subjects requiring larger room and every student cannot take exams in more than 2 subjects in a day. Solving this problem manually, i.e. using human decision making, is highly time-consuming and prone to mistakes such as schedule conflicted issues. This research will apply the grid puzzle, shown in Figure 3, to tackle the described problem.

0

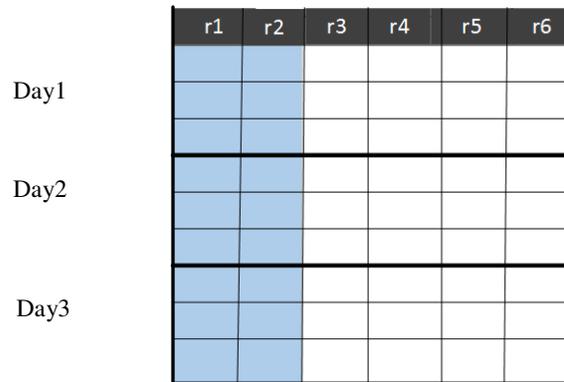


Figure 3. Grid puzzle for examination scheduling problem

From Figure 3, it can be seen that the columns represent rooms or venue of the exam. There are 2 types of rooms in this problem: 1) regular-sized rooms indicated by the white-grids and 2) larger sized rooms indicated by the shaded-grids. Rows of the puzzle represent time slots of the exam. Assuming there are 3 timeslots per day, the thick horizontal lines are used to separate days during the exam period. Thus, Figure 3 is shown that there are 6 rooms available for the exam with 2 large rooms and the exam period lasts 3 days. The objective of the defining game is to assign subject ID to the puzzle such that operational constraints are satisfied. The rules of the game are setup to match the businesses rules of the problem as detailed in Table 1.

Table 1 Business’s and game’s rules of the problem

Business ‘s rules	Game ‘s rules
A. All subjects have to be assigned to the schedule and each subject takes only 1 exam	A. All the numbers indicating subject IDs, can be used only once
B. Students should not take more than 2 exams in a same day	B. In a day sub-region, the number of assigned subjects for each year cannot be over 2
C. Some subjects require large rooms	C The subjects that requires large rooms should be assigned to the given area only

## 5. IMPLEMENTATION

The problems is implemented by using Choco, a Java based CP library. The constraints declared in Section 4 as the rules of this game can be solved by CP as follows:

### 5.1 “All the numbers indicating subject IDs, can be used only once”

Global constraint is a category of constraints that are defined for solving practical problems where association between variables are not limited to ‘local’ consideration [18]. Global constraints are well documented to define 423 constraints in [19]. Global cardinality is a global constraints used to tackle this requirement. The constraint enable limiting the lower bound and upper bound together with the number of times that those values can be used. Imposing the Global cardinality constraint to satisfy this rule in Choco is as the following simplified statement.

```
Impose globalCardinality(S,[0,20],all the number in the range except 0 is only assigned 1 time)
```

The representation for this constraint is depicted in Figure 4. In this application, each variable Subject ID (S) = {1, 2, 3, 4...20) represents a sequence of continuous subject ID. A dummy value 0 is required to indicate that there is no assignment given to that timeslot. Therefore, the domain of this variable, i.e. for 20 subjects, is ranged from [0, 20). The global cardinality is enforced every S, except 0, appearing only once

r1	r2	r3	r4	r5	r6
S1	S2	S3	S4	S5	S6
					Sn

Figure 4. Problem modelling to tackle constraint 5.1

### 5.2 “In a day sub-region, the number of assigned subjects for each year cannot be over 2”

The model of the year of subject is similar the Subject ID as shown in Figure 5. There are four year of students from 1 to 4. However, similar to the previous constraint, a dummy value (0) is required to indicate a ‘no-assignment’. The domain for this variable is therefore ranged from [0, 4].

r1	r2	r3	r4	r5	r6
Y1	Y2	Y3	Y4	Y5	Y6
					Yn

Figure 5. Problem modelling to tackle constraint 5.2

Due to the fact that rows in the puzzle indicate time slot of the exam, Globalcardinality is used to control the number of the domain 1-4 appearing at most twice in each day region. The algorithm for tackling this rules of the defined puzzle is shown in Figure 6.

```

FOR Each day
    Impose GlobalCardinality(Year, [0,4],all the number in the range except 0 is only assigned 2 time)
ENDFOR
    
```

Figure 6. Algorithm for tackling the constraint 5.2

**5.3 “The subjects that requires large-rooms should be assigned to the defined areas only”**

Two larger rooms are defined for the first two columns as shown in Figure 7. Assignment to this area are limited to the subject that required. The subject that require larger room have to be defined in a problem statement, and this value will never be assigned outside that area.

r1	r2	r3	r4	r5	r6	r7	r8

Figure 7 Problem modelling to tackle constraint 5.3

To implement this constraint in Choco, the constraint ‘among’ is applied to limit a subject ID assignment bounded in a predefined list of large rooms. This constraint is only applied to the shaded area of the puzzle. So a constraint is defined within a nested loop. The algorithm is depicted as Figure 7a.

```

FOR i = 0 To LastRow
    For j = To LastColumnLargeRoom
        Impose among (S[i][j], LargeroomList)
    ENDFOR
ENDFOR

```

Figure 7a. Problem modelling to tackle constraint 5.3

#### 5.4” Associating IDs to other attributes”

Being that a grid puzzle is 2D, the limitation in problem modelling is an unknown variable that can be solved one at a time. In practice, there are multiple variables to consider in one problem. For example, the example problem involved with Subject ID and year of the subject. Modelling the problem using a grid puzzle requires to solve the problem separately. The internal constraint beside the explicit constraints of the problem is required to associate with other solving variables. This can be done by imposing constraints to associate variables. In CP, a compatibility between variable can be enforced by declaring a feasible pair i.e. between subject ID and the year variable. This will enable interpretation of which subject is belong to. The algorithm for binding 2 variables is indicated as Figure 8.

```

FOR i = 0 To LastRow
    For j = To LastColumnLargeRoom
        Impose feasiblepair (S[i][j],Yr[i][j],DefinedPair)
    ENDFOR
ENDFOR

```

Figure 8. Algorithm for associating Subject ID with its year

## 6. RESULTS AND DISCUSSION

This Section demonstrates the use of the grid puzzle defined to solve the exam scheduling problem. The sample question is given in Table 2, and brief clarification on the problem is as follows:

Table 2. Requirements of the problem

Subject ID	Year	Large section (yes or no)
1	1	Yes
2	1	No
3	1	No
4	1	No
5	1	No
6	2	No
7	2	Yes
8	2	No
9	2	No
10	2	No
11	3	No
12	3	Yes
13	3	No
14	3	No
15	3	No
16	4	No
17	4	No
18	4	No
19	4	Yes
20	4	No

From Table 2., there are 5 subjects for each year i.e. subject 1-5 for the first year, 6-10 for the second year, 11-15 for the third year, and 16-20 for the fourth year. 5 subjects require larger room: 1, 3, 7, 12, and, 19. Solving this grid puzzle using our proposed method can result in the following scheduling as depicted in Figure 7.

r1	r2	r3	r4	r5	r6
19	12	20	15	10	9
3	1	0	0	0	0
0	0	0	0	0	0
7	0	18	17	14	13
0	0	8	5	4	0
0	0	0	0	0	0
0	0	16	11	6	2
0	0	0	0	0	0
0	0	0	0	0	0

Figure 7. Scheduling result

The result indicates that 3 defined major constraints are satisfied; 1) all subjects are allocated to the schedule 2) there are no more than 2 exams for every year subject and 3) the subject that has larger class-sizes are allocated to the larger room.

In this paper schedule result is generated under CSP focus. Figure 7 show only one possible solution, actually several more possible solutions can be generated. CSP solving does not specify which solution is better than the other, when an optimal solution is required, the problem can be simply expanded to the “Constraint Satisfaction Optimisation Problem (CSOP)” by applying objective function to the model e.g. minimise spanning time.

## 7. CONCLUSION

This paper aims at tackling the problem formulation issue of using CP solving CSP. Applying grid puzzles to represent the problem is an alternative solution to get started solving practical problem. The paper shows the success of using the grid puzzle to solve simple examination scheduling problem. Three operational constraints are addressed; 1) all the subjects are scheduled the exam 2) students can take at most 2 subjects per a days and 3) the schedule allocates the rooms to meet capacity requirement. The future work of this research is to impose more constraint to this problem also applying the model to similar scheduling problems. This work has led to the new research question is the proposed method simple enough for non-computing user? The planned field evaluation is to conduct to evaluation of the proposed method by university administration staff. Subject to success of the proposed method, anyone not limited to computing users who understand the problem can contribute in the problem solving process using CP. In practice, operational workers might be able to formulate a CSP model to cooperate with a Constraint Programmer to shorten problem solving time, or they can even solve the problem by themselves.

## REFERENCES

- [1] Lozano, R.C., Constraint Programming For Random Testing Of A Trading System, In School Of Information And Communication Technolog. 2010, Royal Institute Of Technology: Stockholm, Sweden.
- [2] Bartak, R. Constraint Programming: What Is Behind? In Workshop On Constraint Programming For Decision And Control (Cpdc'99). 1999. Gliwice, Poland.
- [3] Rossi, F., P.V. Beek, And T. Walsh, Introduction, In Handbook Of Constraint Programming, F. Rossi, P.V. Beek, And T. Walsh, Editors. 2006, Elsevier: London.
- [4] Apt, K.R., Principles Of Constraint Programming. 2003, Cambridge ; New York: Cambridge University Press.
- [5] Freuder , E.C. And A.K. Mackworth, Constraint Satisfaction: An Emerging Paradigm, In Handbook Of Constraint Programming, F. Rossi, P.V. Beek, And T. Walsh, Editors. 2006, Elsevier: Oxford.
- [6] Tsang, E., Foundations Of Constraint Satisfaction. 1993, London: Academic.
- [7] Bourdais, S., P. Galinier, And G. Pesant, Hibiscus: A Constraint Programming Application To Staff Scheduling In Health Care. Principles And Practice Of Constraint Programming - Cp 2003, Proceedings, 2003. 2833: P. 153-167.
- [8] Topaloglu, S. And I. Ozkarahan, A Constraint Programming-Based Solution Approach For Medical Resident Scheduling Problems. Computers & Operations Research, 2011. 38(1): P. 246-255.
- [9] Hanset, A., N. Meskens, And D. Duvivier. Using Constraint Programming To Schedule An Operating Theatre. In Health Care Management (Whcm), 2010 Ieee Workshop On. 2010.
- [10] Van Leeuwen, P., H. Hesselink, And J. Rohling, Scheduling Aircraft Using Constraint Satisfaction. Electronic Notes In Theoretical Computer Science, 2002. 76(0): P. 252-268.
- [11] Blomdahl, K.S., P. Flener, And J. Pearson, Contingency Plans For Air Traffic Flow And Capacity Management. 2010
- [12] Valouxis, C. And E. Housos, Constraint Programming Approach For School Timetabling. Computers & Operations Research, 2003. 30(10): P. 1555-1572.
- [13] Abdennadher, S., M. Aly, And M. Edward, Constraint-Based Timetabling System For The German University In Cairo, In Applications Of Declarative Programming And Knowledge Management. 2009, Springer. P. 69-81.
- [14] Celik, M., Et Al., Comparing Asp And Cp On Four Grid Puzzles. 2009.

- [15] Sudoku. [Cited 2014 07 Feb]; Available From: [Http://En.Wikipedia.Org/Wiki/Sudoku](http://En.Wikipedia.Org/Wiki/Sudoku).
- [16] Crossword. [Cited 2014 05 Feb]; Available From: [Http://En.Wikipedia.Org/Wiki/Crossword](http://En.Wikipedia.Org/Wiki/Crossword).
- [17] Eight Queens Puzzle. Available From: [Http://En.Wikipedia.Org/Wiki/Eight\\_Queens\\_Puzzle](http://En.Wikipedia.Org/Wiki/Eight_Queens_Puzzle).
- [18] Bulatov, A.A. And D. Marx, Constraint Satisfaction Problems And Global Cardinality Constraints. Communications Of The Acm, 2010. 53(9): P. 99.
- [19] Beldiceanu, N., M. Carlsson, And J.-X. Rampon, Global Constraint Catalog, (Revision A). 2012.

## **AUTHOR**

***Noppon Choosri*** is a director of the Software, Management and Animation by Radical Technologies (SMART) Research centre, College of Arts, Media and Technologies, Chiang Mai University, Thailand. He is also a lecturer at Software Engineering Department. He received his B.Sc. in Computer Science and M.Sc. in Information Management on Environment and Resources from Mahidol University, Thailand and PhD in Computing Science from Staffordshire University, U.K. His research interest involves applying information technologies to solve practical operational problem in various areas including logistics, knowledge management, tourism, medical science, and environmental studies

