SOLVING SCHEDULING PROBLEMS AS THE PUZZLE GAMES USING CONSTRAINT PROGRAMMING

Noppon Choosri
SMART Research Centre, College of Arts Media and Technology, Chiang Mai University, Chiang Mai, Thailand

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 practical scheduling problems. The examination scheduling and logistic fleet scheduling are presented as operational games. 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; fleet 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 solve[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 := y1, . . ., yk where k > 0, with respective domains D1, . . .,Dk. 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 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 problems. 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 problems, Section 5 discusses the CP implementation, Section 6 discussed the results 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 NxM 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× N chess board, where N ≥ 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 NxN grids in a way that none of the words are used more than once. Sudoku is usually played on 9 x 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 x3 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 FOR SCHEDULING PROBLEM

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 described above.

Figure 2. CP problem solving

Grid puzzles representations, i.e. using 2 Dimension (2D), NxM , 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
scheduling problems in two application; 1) examination scheduling and logistics fleet scheduling which are outlined as follows:

4.1 Examination scheduling problem

The problem 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. The grid puzzle to tackle the described problem is shown as Figure 3.

![Grid puzzle for examination scheduling problem](image)

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

In this problem, fleets of lorries are required to deliver orders of product to customers. Loads can be split in the given time window to finish delivery, also lorries are required to travel together as a fleet. This problem is a typical transport problem for grain suppliers in logistics. In addition, this class of the problem is defined as a truckload problem in which each lorry makes a door-to-door delivery of single product from depot to a customer. The grid puzzle to tackle the described problem is shown as Figure 4.

The leftmost column in shaded is initially given to indicate the number of tonnages required for each order. The objective of the game is to fill the number (indicating tonnage) to satisfy the rules of the game. The major vertical line separates the day of delivery in the given time windows as well as separating fleet of lorries to be assigned. The minor vertical lines are for representing load assignment to each vehicle having size equal the maximum fleet size. Zero assignment to vehicles mean that lorry is not assigned, on the other hand, if non-zero assignments are given to the same fleet, all the vehicles are used for that fleet. The operational constraints are considered as the rules of this game as shown in Table 2.

Table 2 Business’s and game’s rules of the logistic fleet scheduling problem

<table>
<thead>
<tr>
<th>Business rules</th>
<th>Game’s rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Each lorry cannot assign load excess its capacity</td>
<td>1. An valid assignment is a set of positive Integer in the given interval that does not exceed the maximum capacity</td>
</tr>
<tr>
<td>2. Duplicate assignments are not allowed unless lorry return to a depot</td>
<td>2. Re-assignment a non-zero to the same position of column in the delivery period is not allowed</td>
</tr>
<tr>
<td>3. All orders have to be served in exact amount</td>
<td>3. Sum of the assigned value in each row equals the given initial number</td>
</tr>
</tbody>
</table>
5. IMPLEMENTATION

The problems are implemented by using Choco, a Java based CP library. The constraints for examination scheduling and logistic fleet scheduling problems declared in Section 4 as the rules of the games can be solved by CP as follows:

5.1 Examination scheduling problem

Implementation details for the examination scheduling are discussed as follows:

5.1.1 “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]. Globalcardinality 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 Globalcardinality constraint to satisfy this rule in Choco is as the following simplified statement.

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

The representation for this constraint is depicted in Figure 5. 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

![Figure 5. Problem modelling to tackle constraint 5.1.1](image)

5.1.2 “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 6. 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].
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 7.

\textbf{FOR Each day}
\begin{itemize}
  \item Impose GlobalCardinality(Year, \{0,4\}, all the number in the range except 0 is only assigned 2 time)
\end{itemize}
\textbf{ENDFOR}

\textbf{Figure 7. Algorithm for tackling the constraint 5.1.2}

\subsection*{5.1.3 “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 8. Assignments to this area are limited to the subject that required. The subjects that require larger room have to be defined in a problem statement, and this value will never be assigned outside that area.

\textbf{Figure 8. Problem modelling to tackle constraint 5.1.3}

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 9.
5.1.4 “Associating IDs to other attributes”

Being that a grid puzzle is 2 Dimension (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, 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 10.

5.2 Logistic fleet scheduling

In the examination scheduling example, unknown variables, i.e. student ID and year, are initially solved independently and imposing the feasible pair to combine the result. This example demonstrates a game that locates all variables into the puzzle as depicted in Figure 4. Implementation details for the fleet scheduling problem is discussed as follows

5.2.1 “A valid assignment is a set of positive Integer in the given set”

The feasible assignments to lorry are assignments that do not exceed capacities of vehicles which are the assignments should be a number in the set of [0, maximum capacity]. However, to reduce
a solution space, also taking practical operation into account. The feasible assignment for each lorry can be limited to a few number that are usually assigned as operated manually i.e. [lower bound, maximum capacity]. The algorithm for this constraint is depicted in Figure 11.

**Figure 11. Algorithm for tackling the constraint 5.2.1**

### 5.2.2 “Re-assignment a non-zero to the same position of column in the delivery period is not allowed”

Non-zero assignment to the particular column is interpreted amount of loads given to the vehicles. When the number is shown up, it also mean the vehicle is part of the fleet on its service for certain order. Therefore, duplicated assignment during the service time of any vehicle (delivery time + break time) shall not be appeared in a puzzle solution. The algorithm to impose constraint to satisfy this constraint is shown as Figure 12

**Figure 12. Algorithm for tackling the constraint 5.2.2**

### 5.2.3 Sum of the assigned value in each row equals the given initial number

The exact amount of products shall be delivered to meet demand of orders within the commitment windows. As delivery can be split, the total load delivered for each order is a sum of fleets i.e. sum of number in rows. The algorithm is shown as Figure. 13
6. RESULTS

This Section demonstrates the use of the grid puzzle defined to solve the exam scheduling and logistic fleet scheduling problem. Scenarios are used to test the correctness of the method in two application domains as the following details:

6.1 Examination scheduling scenario

The sample question is given in Table 2, and brief clarification on the problem is as follows:

Table 2. Requirements of the examination problem

<table>
<thead>
<tr>
<th>Subject ID</th>
<th>Year</th>
<th>Large section (yes or no)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>No</td>
</tr>
</tbody>
</table>

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 14.
The result indicates that 3 defined major constrains 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.

6.2 Fleet scheduling scenario

A company has received 3 orders from the customer as shown in Table 3 Depot-to-Depot delivery are required to finish the shipping in 5 day. The company has 20 vehicles. Each delivery has duration of 3 days including break times for drivers

Table 3 Requirement of the logistic problem

<table>
<thead>
<tr>
<th>Order number</th>
<th>Amount (tons)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>120</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
</tr>
</tbody>
</table>

The solution of the logistics example is shown as Figure 15. The interpretation is that there are 3 orders to the company which are 70, 120, 100 respectively. The company has 5 vehicles, each delivery consumes 3 days. Three fleets are composted to make this deliveries. For example, the first fleet is composed of vehicle#2-5 starting service on the first day to service the second order (120 tons), each vehicle contributes a load of 30 tons. The vehicles being dispatched will be available to deliver the next service on the 5th day for the first order (70 tons) and the third order (100 tons).

In this paper two schedule results are generated under CSP focus. Figure 7 and Figure 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. Not only formulating problem as puzzles can contribute a simplicity to problem representation, but the method facilitate implementation process also. For example, when problems are formulated as puzzles, typical set of constraints in CP such as Among, Globalcardinality, Equal, etc. can be imposed to the problems the paper shows the success of using the grid puzzle to solve two different simple scheduling problems. In the examination 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. In the logistics fleet scheduling problem, the paper tackles; 1) Each lorry cannot assign load excess its capacity 2) Duplicate assignments are not allowed unless lorry return to a depot and 3) All orders have to be served in exact amount. The future work of this research is to impose more constraint to the 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 operational staffs. 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


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.