Lab Project

CSE – 4618

Automated Factory Management System

for

Delicia Foods Ltd.

Md. Zahidul Islam, 160041010
Md. Sakif Khan, 160041039

Lab Group : **1A**

Introduction

Our system solves three separate yet interconnected problems related to factory management using three Minizinc models.

1. **prodPlan.mzn :** Objective is how much of each product should be produced given a certain fixed budget to buy resources. we are given a fixed budget for raw materials. Also, user will input types and amount of resources. A consumption array will tell us how much of resource each product needs. A profit array signifies how much profit each product makes. The model finds and array which contains how much each type of product needs to produced. This array will be a necessary data for the next model.

2. **prodSchedule.mzn :** Each product goes through five processes - "mixing","moulding","baking","decorating","packaging". Each of these processes take different amount of time for different products stored in the 2D array "duration". For a certain product, mixing needs to take place before moulding, moulding needs to take place before baking and so on. There are ensure using precendence. The global function "cumulative" finds out the start time and duration of each of the process and the total span.

3. **prodWorkerAssign.mzn :** There are some workers in the factory. Let's say, 24. Each of these workers need to be assigned to some process (mixing, baking etc.). They have different ability in each of the process stored in the 2D array "ability". The workers cannot be assigned to the same process for consecutively three days as this will make sure the workers don't get bored. The model outputs the assignment of each person to a certain process for 1 to n days.

The models and data files are stored in the folder "scripts". The folder also contains the python GUI scripts for this system.

P.T.O.

Requirements

1. Minizinc
2. Python 3
3. Python libraries –
    a) Pymzn
    b) PyQT5
    c) sys

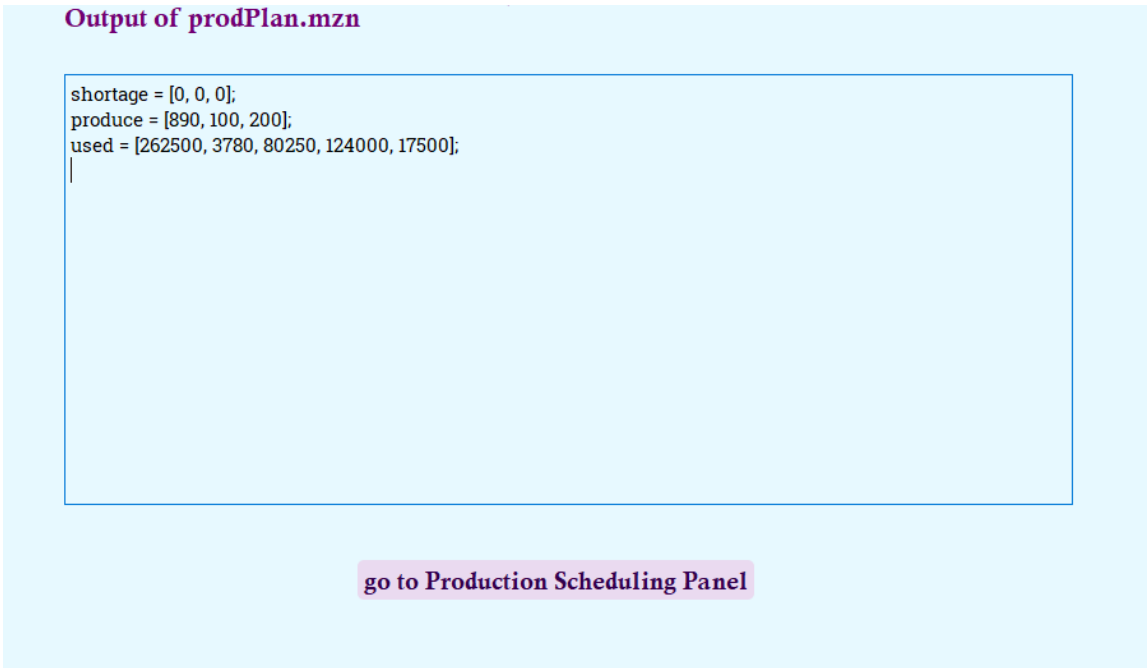
How To Run

Running with Python :

- Open a terminal
- Go to the folder “scripts”.
- Run mainWin.py - “python mainWin.py”.
- Click on “manufacturing panel”.

- It will open up the prodPlan window. Click on attach a dzn file and choose the file **"prodPlan.dzn".**

- dzn file can also be provided by typing in the textbox displayed above. But, in this case care should be taken so that the dzn data entered in the textbox are valid.
- After choosing the dzn file, click on "done!". Wait for a bit, the minizinc will solve the problem and send solutions to python.
- The output looks like this.



- Click on "go to Production Scheduling Panel".

- Now run Production Scheduling in the same way as Production Planning was done. The dzn file/textbox should not contain “produce” array. As it was calculated in the previous and will be passed automatically to the next model using python.
- Choose the file “prodSchedule.dzn” as the dzn file.

- Click on “go to Worker Assignment panel”.
- This is the third and final task of our system. Which is assigning workers to different tasks.
- Click on “attach a dzn file” and select **“prodWorkerAssign.dzn”.**
- The output for this third model is actually a demo. Because, the minizinc takes very long to solve the problem. To actually run the third model, use minizinc IDE.

set data for prodWorkerAssign.mzn

```
nprocesses = ;
processName = [ ];

nworkers =  ;
workerName = [ ];
% below is an array (numberOfWorkers x numberOfProcesses)
ability = [ ];

% finds an assignment for next 15 days
nDays =  ;

% minimum worker needed for each process
minimumWorkers = [ ];
```

Or, attach a dzn file G:/books/6TH SEMESTER/AI lab 4618/Project/Delicia Food Factory Managemen

done!

- The demo output of the **“prodWorkerAssign.mzn”** is displayed.

Output of prodWorkerAssign.mzn

```
[format : process => day]

"Peter Quill"
4 => 1,  5 => 2,  5 => 3,  4 => 4,  5 => 5,  1 => 6,  4 => 7,  4 => 8,  5 => 9,  4 => 10,
"Bucky Barnes"
4 => 1,  5 => 2,  5 => 3,  4 => 4,  5 => 5,  5 => 6,  4 => 7,  5 => 8,  5 => 9,  4 => 10,
"Stephen Strange"
4 => 1,  4 => 2,  1 => 3,  4 => 4,  4 => 5,  5 => 6,  4 => 7,  5 => 8,  4 => 9,  4 => 10,
"Shuri"
1 => 1,  4 => 2,  4 => 3,  3 => 4,  4 => 5,  4 => 6,  3 => 7,  4 => 8,  4 => 9,  3 => 10,
"Sam Wilson"
1 => 1,  4 => 2,  4 => 3,  5 => 4,  4 => 5,  4 => 6,  5 => 7,  4 => 8,  4 => 9,  5 => 10,
"Steve Rogers"
3 => 1,  4 => 2,  4 => 3,  3 => 4,  4 => 5,  4 => 6,  1 => 7,  4 => 8,  4 => 9,  1 => 10,
"Wanda Maximoff"
```

OK

- To run the models without using python GUI driver, follow the next instructions.

Running without Python :

- Open Minizinc IDE.
- Open the files below in the IDE.

  - ✓ prodPlan.mzn
  - ✓ prodPlan.dzn

  - ✓ prodSchedule.mzn
  - ✓ prodSchedule.dzn

  - ✓ prodWorkerAssign.mzn
  - ✓ prodWorkerAssign.dzn

- First, run the model “prodPlan.dzn”
- The output will contain the array “produce”. Insert the array by typing in the dzn file, “prodSchedule.dzn”.
- Now run the “prodSchedule.mzn.”
- Then run the “prodWorkerAssign.mzn.”

Concluding Remarks :

- This project is just a demonstration how constraint satisfaction and discrete optimization models can be applied in business or factory management systems. A lot of works need to be done to make this system applicable in a real life scenario.

_____X_____