Project Name: Meal Planner

Yifei Dong – dongyife

key responsibilities: preliminary research(minor), encode program engine(major), explore propagation techniques(minor), testing(minor)

Yuwen Huang – huang123

key responsibilities: preliminary research(minor), encode the problem as a CSP model(major), explore ordering techniques(minor), testing(minor)

Type of Project: CSP

Project Motivation/Background:
Most of the meal plans provided online are fixed meal plans. We were thinking about making a meal planner which makes customized meal plans for users. The meal planner will ask for the user's age, gender, height, weight and exercise level to generate a suggested calories intake for a day. The user can also type in what he wants to eat for breakfast/lunch/dinner for each day of week specifically.

We are implementing this application using CSP model. The list of inputs we collect from users can be coded as constrains in our CSP model. CSP is suitable for this problem because rather than finding a path to reach to the suitable meal plans, we only need the list of final results to be represented to users. Moreover, finding the shortest path in the search space to achieve at the final meal plan is not one of our concerns. Both facts stated above make search algorithms such as BSF, DSF, A* unsuitable. Furthermore, game tree search would not be suitable in our case, because meal planner is not a two-person game where each player take turn to make moves.

Methods:
The CSP employs backtracking as the algorithm to find the solution. Also, we have two propagators, which are forward checking and generalized arc consistency. The forwarding checking algorithm checks constraints with only one uninstantiated variable, all of its values that violates the constraints will be pruned. The generalized arc consistency algorithm checks if a variable is GAC and enforce GAC if it is not.

All the information of food is implemented as a dictionary in mealplan.py with the food name as the key of dictionary. Take health into consideration, we arranged the basic structure of meal plans to make sure the user can have enough nutrition in each meal. The following is the basic structure:
Breakfast: one beverage, one meat/diary, one vegetable, one fruit
Lunch: one beverage, one meat/seafood, one main, one vegetable
Dinner: one meat/seafood, one main, one vegetable, one fruit

We are making a 2D list to represent the meal plan. The list consists of seven lists, which each of them represents a day of the week. The first list represents Monday, whereas the last one represents Sunday. The inner list is separated into 12 variables. The first four variables are for breakfast, the second four variables are for lunch and the last four variables are for dinner. Also, we have restricted the food category for each variable domain, so the basic structure of the meal plan is fixed. The sequence of the type of the variables are the same as the structure listed above.

We have implemented 56 constraints for the one-week meal plan model. The first sevens constraints ensure that the breakfast calorie intake each day is below the limit and each meal includes the required food inputted by users. The second and third set of constrains were created for the similar purpose, but they are enforced on lunch and dinner. The last four sets of constrains validates whether the same food has been planned more than once in a day. Each constraint is specific for one specific category and meal (e.g. one constraint would be created for the fruit category, which will validate whether the different fruit is planned for breakfast and dinner).

Evaluation and Results:

1. Evaluate model with and without heuristic

The following table represents the runtime of the model with and without MRVvar heuristic in seconds. We were using one-week meal plan and GAC propagator for this table.

Constraints	With MRVvar heuristic (s)	Without MRVvar heuristic (s)
Intake calories for breakfast/lunch/dinner: 600, 600, 600 Mind eating the same food twice or more in a day? Yes	3.3813712119999995	3.1848146360000005
Intake calories for breakfast/lunch/dinner: 600, 600, 600 The food that allergic to: shrimp The food that you want to eat for breakfast: egg (days: 1,3,5,6) The food that you want to eat for dinner: apple (days: 2, 4, 5) Mind eating the same food twice or more in a day? No	0.8616094199999997	0.7699605959999998
Intake calories for breakfast/lunch/dinner: 600, 600, 600 The food that allergic to: shrimp The food that you don't want to eat for breakfast: banana (days: 3,4,7) The food that you don't want to eat for lunch: beef (days: 1, 3, 5) The food that you want to eat for breakfast: egg (days: 1,3,5,6), milk (days: 1,2,5,7) The food that you want to eat for dinner: apple (days: 2,4,5) Mind eating the same food twice or more in a day? Yes	1.7473753040000002	2.5346681479999997

From the table above, we noticed that when there aren't too many constraints, the runtimes of the model with and without MRVvar heuristic are similar. But when there are more than a few constraints, the model runs significantly faster with MRVvar heuristic than without MRVvar heuristic.

2. Evaluate the two different propagators, see the time difference

The following table represents the runtime of the model with FF and GAC propagators in seconds. We were using one-week meal plan and MRVvar heuristic for this table.

Constraints	FC (forward checking) (s)	GAC (generalized arc consistency) (s)
Intake calories for breakfast/lunch/dinner: 600, 600, 600 Mind eating the same food twice or more in a day? Yes	More than 10 minutes	3.339952598
Intake calories for breakfast/lunch/dinner: 600, 600, 600 The food that allergic to: shrimp The food that you want to eat for breakfast: egg (days:	More than 10 minutes	0.8616094199999997

1,3,5,6) The food that you want to eat for dinner: apple (days: 2, 4, 5) Mind eating the same food twice or more in a day? No		

From the table above, we can see that the FC propagation is taking a much longer time than GAC propagator no matter how many constraints are there.

3. Evaluate the meal plan's variability. We are evaluating this using GAC propagator and MRVvar heuristic.
Constraints:
Intake calories for breakfast, lunch, dinner: 600, 800, 700
The food that allergic to: shrimp
The food that you want to eat for breakfast: egg (days: 1,3,5,6)
The food that you want to eat for dinner: apple (days: 2, 4, 5)
Mind eating the same food twice or more in a day? Yes

Result:
Search made 84 variable assignments and pruned 520 variable values
Solution

Monday:
Breakfast:
1 cup of orange juice
1 each of egg
2 serving of artichoke
1 serving of mango
Lunch:
1 cup of apple juice
1 serving of beef
100 g of potatoes
2 serving of squash
Dinner:
200 g of duck
1 serving of rice
1 serving of turnip
1 serving of pear

Tuesday:
Breakfast:
1 cup of apple juice
100 g of turkey
1.0 serving of bok choy
4 ring of pineapple
Lunch:
2 serving of milk
100 g of pork
2 serving of rice
1 cup of beans
Dinner:
2 each of egg
200 g of potatoes
1 serving of mushrooms
2 serving of apple

Wednesday:
Breakfast:
1 serving of milk
2 each of egg
2 serving of turnip
1 serving of melon
Lunch:
2 cup of apple juice
100 g of chicken
2 serving of rice
2 serving of mushrooms
Dinner:
200 g of lamb
200 g of potatoes
3 stalk of celery
1 serving of mango

Thursday:
Breakfast:
2 serving of milk
100 g of turkey
2 cup of beans
2 ring of pineapple
Lunch:
2 cup of apple juice
100 g of chicken
1 serving of rice
2 serving of turnip
Dinner:
100 g of pork
100 g of potatoes
3 stalk of celery
1 serving of apple

Friday:
Breakfast:
2 cup of apple juice
1 each of egg
2 each of cucumbers
2 serving of grapefruit
Lunch:
2 cup of orange juice
100 g of pork
1 serving of rice

1 serving of mushrooms
Dinner:
100 g of duck
200 g of potatoes
0.2 pound of green pepper
2 serving of apple

Saturday:
Breakfast:
1 cup of apple juice
1 each of egg
1.0 serving of bok choy
100 g of raspberries
Lunch:
2 cup of orange juice
200 g of chicken
200 g of potatoes
1 serving of carrots
Dinner:
1 serving of beef
2 serving of rice
1 cup of onions
1 serving of tomato

Sunday:
Breakfast:
1 cup of orange juice
2 serving of beef
2 serving of cauliflower
2 serving of cranberries
Lunch:
2 serving of milk
200 g of chicken
1 serving of rice
1 serving of peas
Dinner:
1 each of egg
200 g of potatoes
0.4 pound of green pepper
2 serving of mango

The result is variant from day to day and the meal plan also satisfies all the constraints. Therefore, the meal planner is working properly.

4. Test whether the generated one-day meal plan is within budget. We are evaluating this using GAC propagator and MRVvar heuristic.
Constraints:
Intake calories for breakfast, lunch, dinner: 600, 800, 700
The food that allergic to: shrimp
The food that you want to eat for breakfast: egg
The food that you want to eat for lunch: apple
Mind eating the same food twice or more in a day? Yes
What is your budget for breakfast? 5
What is your budget for lunch? 10
What is your budget for dinner? 8

Result:
Search made 12 variable assignments and pruned 110 variable values
Solution

Breakfast:	Lunch:	Dinner:
1 cup of apple juice	1 serving of milk	100 g of goose
1 each of egg	1 serving of beef	1 serving of rice
1 serving of peas	100 g of potatoes	1 each of cucumbers
1 serving of grapes	1 serving of okra	1 serving of apple

We evaluated the one-day meal plan calculating the price of each meal using the data from database (mealplan.py) and see if the price satisfies the budgets. The results show the prices are all under budget and the constraints are also all satisfied. Therefore, the one-day meal plan is working properly.

Limitations/Obstacles:

One of the obstacles we faced was that the meal planner gives the same meal plans for every single day in the week. We realized this was happening because we were using the same propagator to generator result for everyday and the sequence of the variables in the domain were always the same. In order to fix this problem, we first tried to shuffle the sequence of the satisfying tuples, but the meal planer is still producing the same meal plans. Then we figured that it is the sequence of domain which should be shuffled at the beginning. After we shuffled the domain of all possible variables, the meal planner is finally giving different meal plans for every day.

Another obstacle we encountered is that the model takes a long time to run. We first decreased the variety of food in our database. The model got faster but the improvement is very tiny. Then we changed the data structure for lunch a bit. We were first providing one beverage, one meat/seafood, one main, one vegetable and one fruit for lunch, we decided not to provide fruit anymore. After this small change, the runtime of the model is shorter.

Furthermore, we also faced another problem we were testing the model. When the user enters a food, which is not in our database for the question s“Please enter the food that you want to eat for breakfast/lunch/dinner”, the meal planner cannot give a solution. Therefore, we added a few codes to tell the user that the food is not in our database and will not be used to build up the meal plan. Moreover, the meal planner only provides certain category of food for each meal, for example, if the user wants to eat fruit for lunch, a solution will not be generated. For this problem, we wrote a readme file which tells the user how this meal planner works and what category of food they can choose for each meal.

The limitation of the meal planner is that when more constraints are entered by the user, the runtime of forward checking increases significantly.

Conclusions:

Building up this meal planner is an interesting and rewarding experience. We learned how to implement a CSP model for a real-world problem. We also learned how to modify the different constraints to make the program run faster, for example, we combined some of the constraints and wrote the constraint which pruned the most variables in the domain at the beginning. Moreover, we figured the way to establish variables (data structure) for a meal planner. Additionally, from testing the model, we gained more knowledge of forward checking and GAC propagation. The efficiency of GAC is much better than FC under this situation.

For further improvements of the meal planner, we were thinking about adding protein as another aspect of what food to choose for the user. We were also considering adding more meal plan options like lose weight meal plan and children's meal plan. Including more food in the database to make the meal plan more variant is also something we were thinking about.

References:

1. https://caloriebee.com/diets/food-calorie-chart
2. http://bmi-calories.com/calorie-intake-calculator.html