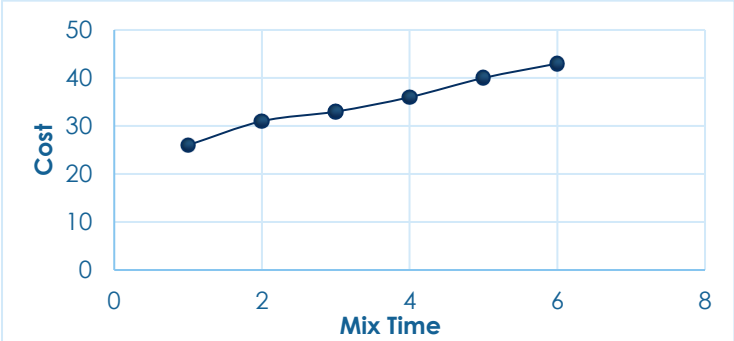
Introduction

This is an analysis of the cake bakery model. It describes the results of running problem files on the domain using Optic planner for windows.

Analysis

Action durations

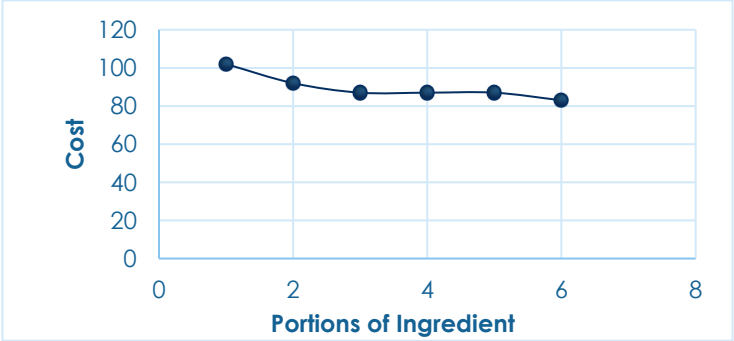
The domain allows for the duration of various actions to be determined in the problem file e.g. the time required to bake a cake in a particular oven. The graph below demonstrates the effect of altering the mix time for a problem. The mix time was set the same for all chefs.



There is a clear positive correlation between the mix time set and the cost of the solution found by the planner. This shows that as one would expect increasing the duration of this action increases the cost of the solution. Similar relations would be found for other action durations. This applies well to a real-life situation as if a chef is slower at mixing in ingredients the time taken to bake the cake would increase.

Portions of ingredients

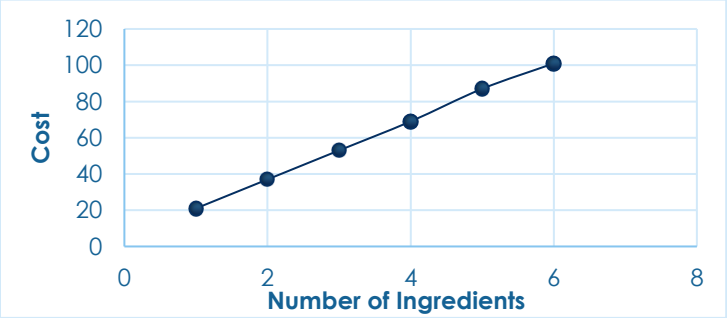
The domain allows for the maximum, required and initial values of the energy of a chef and the amount of each ingredient to be set. The graph below demonstrates the effect of altering the number of portions of each ingredient initially available. The number of portions was set the same for all ingredients and the maximum amount was set equal to the initial. The data was taken from a problem that required the decorating of six cakes.



Overall there is a negative correlation between the number of portions and the cost of the solution found by the planner. This shows that as one would expect increasing the number of portions initially available and available after a shopping trip decreases the cost as chefs are required to go shopping less often. The costs for three, four and five portions are unchanging. This is because the number of trips required to bake six cakes is the same. A similar relation would be found for the energy of the chefs.

Number of ingredients

The domain allows for any number of ingredients to be included in the cakes. All ingredients must be included in each cake. The graph below demonstrates the effect of altering the number of ingredients that must be added.



There is a clear positive correlation between the number of ingredients and the cost of the solution found by the planner. This shows that as one would expect increasing the number of ingredients that must be added increases the cost of the solution. This applies well to a real-life situation as more complex recipes require more time to combine the ingredients.

Number of chefs

The domain allows for the any number of chefs to work on the cakes. The graph below demonstrates the effect of altering the number of chefs.

At low values, an increase in the number of chefs has a significant negative effect on the cost but after a certain number, three in this example, further increases have no effect. This applies well to a real-life situation as more hands will speed up the work as long as there is work for them to do. Having too many chefs is not efficient. Similar relations would be found for the number of utensils.

To conclude, the model accurately describes a cake bakery within the limitations of PDDL modelling. A working cake bakery will of course have more variables that must be considered but this could be used for rough estimates.

Appendix

Problem 1

```
(define (problem bake-a-cake-1)
(:domain bake-a-cake)
(:objects
sugar egg butter chocolate - ingredient
chef1 chef2 - chef
c1 c2 - cake
spoon1 spoon2 - spoon
bowl1 bowl2 - bowl
tray1 - tray
oven1 - oven
oven2 - oven
)

(:init
(available spoon1)
(available spoon2)
(available bowl1)
(available bowl2)
(available tray1)
(available oven1)
(available oven2)
(available c1)
(available c2)
(available chef1)
(available chef2)
(clean spoon1)
(clean spoon2)

(= (bake-time oven1) 3)
(= (bake-time oven2) 2)
(= (energy chef1) 25)
(= (decorate-time chef1) 2)
```

```
(= (mix-time chef1) 2)
(= (wash-time chef1) 3)
(= (travel-time chef1) 2)
(= (energy chef2) 10)
(= (decorate-time chef2) 4)
(= (mix-time chef2) 2)
(= (travel-time chef2) 1)
(= (wash-time chef2) 1)
(= (amount sugar) 5)
(= (amount egg) 5)
(= (amount butter) 2)
(= (amount chocolate) 2)
(= (required-amount) 5)
(= (max-amount) 5)
(= (max-energy) 30)
(= (required-energy) 5)
(= (rest-time) 5)
)

(:goal
(and
(decorated c1)
(decorated c2)
))

(:metric minimize (total-time))
)
```

Problem 2

```
(define (problem bake-a-cake-2)

(:domain bake-a-cake)

(:objects
sugar egg butter chocolate vanilla - ingredient
chef1 chef2 chef3 - chef
c1 c2 c3 c4 - cake
spoon1 spoon2 - spoon
bowl1 - bowl
tray1 tray2 - tray
oven1 - oven
oven2 - oven
)

(:init
(available spoon1)
(available spoon2)
(available bowl1)
(available tray1)
(available tray2)
(available oven1)
(available oven2)
(available c1)
(available c2)
(available c3)
(available c4)
(available chef1)
(available chef2)
(available chef3)
(clean spoon1)
(clean spoon2)
```

```
(= (bake-time oven1) 3)
(= (bake-time oven2) 2)
(= (energy chef1) 25)
(= (decorate-time chef1) 2)
(= (mix-time chef1) 2)
(= (wash-time chef1) 3)
(= (travel-time chef1) 2)
(= (energy chef2) 10)
(= (decorate-time chef2) 4)
(= (mix-time chef2) 2)
(= (travel-time chef2) 1)
(= (wash-time chef2) 1)
(= (energy chef3) 15)
(= (decorate-time chef3) 5)
(= (mix-time chef3) 4)
(= (travel-time chef3) 1)
(= (wash-time chef3) 2)
(= (amount sugar) 5)
(= (amount egg) 5)
(= (amount butter) 2)
(= (amount chocolate) 2)
(= (amount vanilla) 10)
(= (required-amount) 5)
(= (max-amount) 10)
(= (max-energy) 30)
(= (required-energy) 5)
(= (rest-time) 5)
)

(:goal
(and
(decorated c1)
```

```
(decorated c2)
(baked c3)
(baked c4)
))

(:metric minimize (total-time))
)
```

Problem 3

```
(define (problem bake-a-cake-2)

(:domain bake-a-cake)

(:objects
sugar egg butter chocolate vanilla strawberry - ingredient
chef1 chef2 chef3 chef4 - chef
c1 c2 c3 c4 - cake
spoon1 spoon2 - spoon
bowl1 bowl2 bowl3 - bowl
tray1 tray2 - tray
oven1 - oven
oven2 - oven
)

(:init
(available spoon1)
(available spoon2)
(available bowl1)
(available bowl2)
(available bowl3)
(available tray1)
(available tray2)
(available oven1)
(available oven2)
(available c1)
(available c2)
(available c3)
(available c4)
(available chef1)
(available chef2)
(available chef3)
(available chef4)
```

(clean spoon1)
(clean spoon2)

(= (bake-time oven1) 3)
(= (bake-time oven2) 2)
(= (energy chef1) 25)
(= (decorate-time chef1) 2)
(= (mix-time chef1) 2)
(= (wash-time chef1) 3)
(= (travel-time chef1) 2)
(= (energy chef2) 10)
(= (decorate-time chef2) 4)
(= (mix-time chef2) 2)
(= (travel-time chef2) 1)
(= (wash-time chef2) 1)
(= (energy chef3) 15)
(= (decorate-time chef3) 5)
(= (mix-time chef3) 4)
(= (travel-time chef3) 1)
(= (wash-time chef3) 2)
(= (energy chef4) 30)
(= (decorate-time chef4) 3)
(= (mix-time chef4) 2)
(= (travel-time chef4) 2)
(= (wash-time chef4) 3)

(= (amount sugar) 5)
(= (amount egg) 5)
(= (amount butter) 2)
(= (amount chocolate) 2)
(= (amount vanilla) 10)
(= (amount strawberry) 10)

```
(= (required-amount) 5)
(= (max-amount) 10)
(= (max-energy) 30)
(= (required-energy) 5)
(= (rest-time) 15)
)



(:goal
(and
(decorated c1)
(decorated c2)
(baked c3)
(baked c4)
))

(:metric minimize (total-time))
)
```