**4CCS1IAI – Introduction to Artificial Intelligence – Coursework 2015**

Group Members:

* Tahmidul Islam (KNumber)
* Hassan Minhas (KNumber)
* Muhammad Rohman (K1461540)

Introduction

Our domain is a model of pizza delivery, inspired by the logistics domain.

Food delivery is a crucial service for many takeaways in the industry, especially near central London as the market is very competitive. Quality of delivery service is an important factor in retaining customers. Fast, consistent delivery can result in more regular customers. However, slow and inconsistent delivery services can result in customers deeming the business unreliable. There is then the possibility of negative reviews.

The purpose of our domain and problem files is to allow the user to plan a route for delivering a set number of meals to locations in the most efficient way possible, in terms of time and fuel. The domain takes into account factors such as delivery destinations, routes, and delivery vehicles. It measures the time it takes to perform each action. It takes into account the fuel of each vehicle when calculating the solution. Furthermore this model can be easily changed, in the future, for any other delivery/logistics problems.

Domain Relaxations and Limitations

During the creation of this domain we came up with a number of assumptions of this system. One assumption is that the time it take it takes between routes does not change (meaning it will not take in to account of any traffic or bad weather conditions). However the user can manually change the time between routes in the problem file if he wishes.

Furthermore, the planner is limited in that it can only take one set of deliveries at a time. In reality, the business could have a constant stream of deliveries. A plan could be produced for the current set of deliveries. However, there could also be more orders, thus making the current plan obsolete. The situation of the business could change within minutes, especially at peak-time.

Another assumption that we made with our domain is that the set of pizzas described in the problem are cooked and ready to be delivered. Our domain does not cover pizza baking. It is strictly limited to delivery.

Results

Evaluation

Appendix

(Pizza Domain, problem file)

**Report Plan**

Intro –

Our domain

* Briefly explain idea behind project
  + ~~Pizza delivery is a way of life~~
  + ~~In London, delivery has become a crucial aspect of restaurant/takeaways.~~
  + ~~Competitive~~
  + ~~Delivery can take up a lot of resources~~
  + ~~Improve delivery service~~
  + ~~This planner could be used to produce the most efficient solution for delivering pizzas (or any other type of food) to customers~~
* ~~Similar to Maps application, but takes into account factors specific to the domain. ( vehicles, fuel, etc)~~
* Talk about problem relaxations and limitations of domain
  + What we don’t take into account and why
    - ~~Pizza baking – In our domain, we assume that the pizzas have already been made. We do not take into account the cooking time of the pizza. The domain is strictly limited to delivery (more on this in the evaluation, see below)~~
    - ~~The planner is limited in that it can only take one set of deliveries at a time. In reality, the business could have a constant stream of deliveries. A plan could be produced for the current set of deliveries. However, there could also be more orders, thus making the current plan obsolete. The situation of the business could change within minutes, especially at peak-time.~~
* Explain choice of planner
  + We tested the domain and planner using Optic and JavaFF
  + Say what type of search the planner uses
  + We used to two planners so that we could thoroughly test the domain and problems. We wanted to see how consistent the performance is.

Domain -

* Briefly describe types, functions, durative-actions
  + Why we used durative actions
    - The key concept of the domain is to find approximate timings for delivering  sets of pizzas to specific areas
    - We made the main subtasks of pizza deliveries as durative actions so that the planner takes these subtasks into account when calculating time

Results -

* Discuss results of several problem files of varying sizes (table, chart, etc)
* Illustrate the time of the planner against the size of the problem
* use graphs when possible
* Compare times of the same problem between Optic and JavaFF?

Evaluation

* Issues during the project
  + To add action for baking the actual pizza
  + One issue that we found with this is that it would be too big to implement the baking side of the domain.
  + For this project, we wanted to focus on the logistics side of pizza delivery.
* Points of interest of the domain
* How to expand the domain in the future
  + We mentioned earlier about an issue we had with the domain, and deciding if we should add pizza baking into the domain.
  + This would be worth exploring in the future.
  + We could make it so that ‘orders’ are delivered to the customer instead of just pizzas. These order object could consist of several other objects such as pizzas, sides, etc. These could each have durative actions.

Appendix