**20**

15

**Pizza Delivery**

Introduction to Artificial Intelligence Coursework 2015

Group Members:

- Tahmidul Islam (K1462015)

- Hassan Minhas (K146907)

- Muhammad Rohman (K1461540)

*We declare that we understand the nature of plagiarism as defined in the Undergraduate Handbook and that the content of this coursework submission is entirely our own work.*

08

**Fall**

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

**Introduction**

Currently the 2nd most popular takeout food is pizza, and over the past 5 years it has surpassed the growth rate of all other food services. An example of a successful pizza parlour would be Pizza Hut, with over 12,583 total restaurants within the US and 90 other countries. A pivotal key to its success lies in their ability to make pizza, at great quality but also being able to deliver the pizza quickly.

It is this that gave us our inspiration to model the delivery aspect of a pizza company. 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. Also, since drivers working for a pizza parlour would have orders fairly regularly, it is very important to keep fuel-cost as low as possible in order to maintain a higher profit margin.

The purpose of our domain and problem files is to allow the user to plan a route for delivering a set number of pizzas to locations in the most efficient way possible, whilst taking into account time, fuel and the fact that there may be more then one possible vehicle transporting the pizza, and therefore our plan allows us to make efficient use of this fact.

Furthermore since no two orders will be the same, we have taken into account that our domain needs to be domain-independent and therefore able to work with multiple types of problems involving different vehicles, number of streets, number of pizza orders etc. We have therefore run multiple tests using different problem files to check a valid plan is still produced.

**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.

Moreover, we always assume that there will be a parking space near the delivery location.

**Results**

The planner we used for our domain was OPTIC. This planner uses A\* search and also is a “Temporal planner where plan cost is determined by preference of time”. Since time was something that was very important to our plan, we believe that this planner would be appropriate to use.

Domain File

LOL

LOL

LOL

LOL





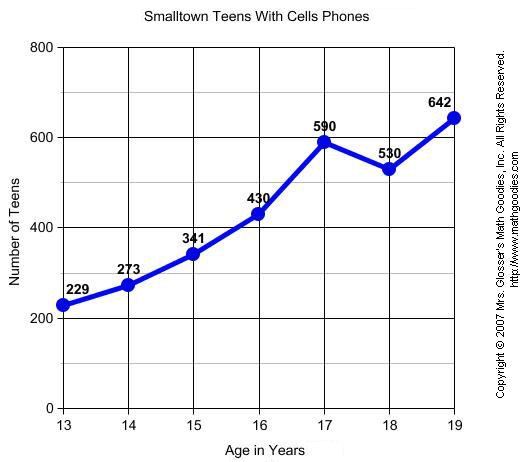




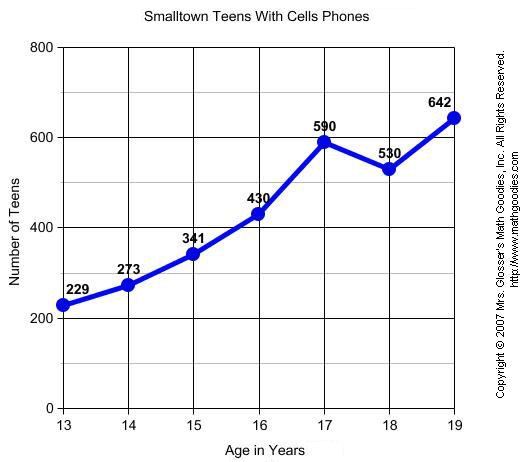




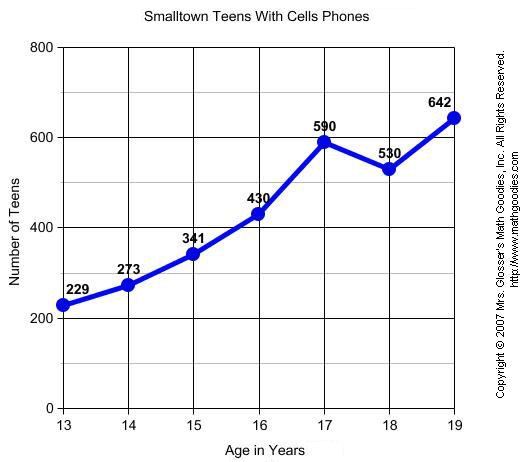
**Analytics**



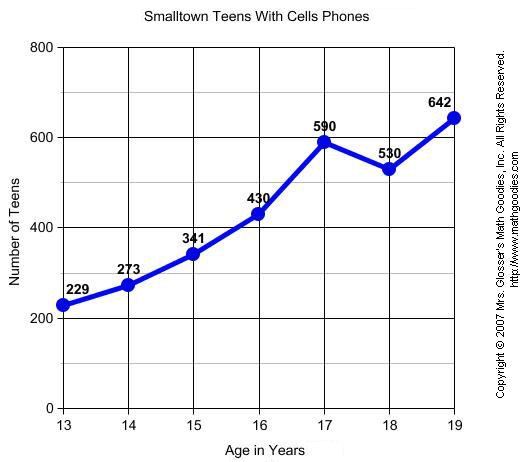
LOL



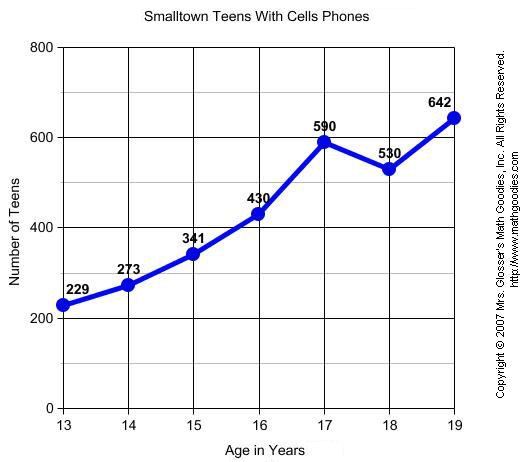
LOL



LOL



LOL



LOL

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~~
  + Say what type of search the planner uses

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