# CAP 4680/6685 EXPERT SYSTEMS
# Spring 2014
# Semester Project

## REPORT

**NAME: MONICA MAVOORI**
**UFID: 46309228**
**EMAIL ID: mmavoori@ufl.edu**

## DESIGN PHILOSOPHY:

For a package delivery system it is very important for the truck to choose the shortest distance it can travel to deliver a package to its destination with in the expected delivery time. This functionality is automated according to the map of the particular state and cities involved. Fact shortest_path with its slots city1, city2, time and via stores this automated values where city1 is the start city to where the package has been arrived and city2 is the destination city to where the packet must be delivered, time is the minimum amount of time required to travel to reach the destination, and via includes the in-between path from start city to destination city storing all the cities a truck must travel to reach the destination. Each and every possible combination of the cities is considered and their shortest path is calculated accurately. Shortest distance from a city to itself is noted as zero.

To implement this design, a clock must be maintained and updated time to time according to the truck move and package arrival. Reached time is maintained for every truck which states the time required to reach the next destination. So according to the truck reach time and package arrival time the clock is incremented. Before incrementing the clock check that there is no package arrived already also there must be no truck whose reach time is less than the updated time.

**Truck to Load Package:**

Whenever a new package arrives following conditions must be satisfied to make sure it has to be picked by proper truck

- There must be a package whose status is arrived and arrival time is equal to the current clock, then a truck which can reach the package early having capacity to fill the package is assigned to pick up the package.
- There must be no package whose arrival time is less than the package which is being picked up
- There may be few situations where two packages has arrived at the same time to same place, so a truck having sufficient capacity to pick them up will reach the package arrival city in same time, then the package whose expected delivery time is earlier is picked up.

Modifications to do:

- When a package is picked up by the truck, it has to wait for the amount of time taken for the truck to come and pick it up. Hence its waiting time is updated, also status changes from “arrived” to “waiting to pick up”.
- When a truck loads a package, its available space must be reduced by the amount of package size; busy time, non-delivery time are updated by the amount of time taken to reach the packages depart city as its being busy but it’s not the time meant for delivering the package. Reached time of the truck is updated, also the package number which it is going to pick up is noted in the truck’s slot ‘numofpack’.

**Package to Queue:**

If a new package arrives and there might be few situations where there is no truck idle to pick it up, and there might be an idle truck but available space is not enough to pick up the package. In such case the package enters into the queue and waits for the truck to pick it up

Modifications to do:

- Modify queue by adding the package number to the beginning of the queue.
- Modify package status as "waiting in queue".

**Queue to Truck:**

- Whenever a package arrives always it checks whether queue is not empty. If queue is not empty and a truck is idle at that clock time then top package from the queue is assigned to that truck which is nearest and new package enters the queue.
- Also, whenever queue is not empty and a truck becomes idle with sufficient amount of space to pick it up, then package from queue is assigned to the truck to deliver it to its delivery city.
- If two trucks waiting in the queue whose arrival time is same, then the package with earlier expected delivery time is assigned to the truck which can reach its depart city early.

Modifications to do:

- Update the queue, by removing appropriate package.
- Modify truck by storing the package number it is carrying, reached time by the time it takes to reach the package depart city. Also its status from "idle" to "going to pick up"; waiting time, busy time and non-delivery time.
- Modify package waiting time, by the time spent in the queue waiting for truck to pick up.

**Truck with Package:**

- Truck after reaching the package depart city, picks it up and will be on its destination to next city on the way. There may be a via city through which the truck must be going to deliver package. Also if there is no such via city it directly delivers the truck to its deliver city.
- If there exists a via city then the truck must go it first and next via city and so to check whether there is a package on its way to pick up. If there is any suck package then the truck picks that too if truck has enough space to carry another one.

Modifications to do:

- The package status is updated to "in-transit" also pick up time, time at which truck picked up the package.
- The truck destination city will be the via city through which it must travel to deliver the package. It's reached time, busy time, delivery time are updated with the time taken to travel to that via city. If no such via city exists all times are updated with the minimum time taken to reach destination city.

**Load Other Packages on-the-way:**

When the truck is travelling from via city if it encounters any package arrived whose depart city is same as the via city and arrival time is same as the truck's reached time then if the truck has enough space to pick up that package, it picks up the package.

Modifications to do:

- If the new package deliver city is on the way, then the trucks destination city is updated as first to reach new package's deliver city and next old package's delivery city.
- If new package deliver city is after the old one's then truck's destination will be to deliver new package after old one.
- The new picked up package number is added to the truck's list of packages carrying.
- Its time's are updated accordingly, to deliver both the packages one after the other.

**Truck Deliver Packages:**

- A truck carrying a package whose destination city is same as the packages destination city then the truck is ready to deliver the package.
- There may be a situation when there are no more packages arriving, so the clock won't be updated, but there is a truck yet to deliver a package. Then the packages are delivered and the extra time taken is noted to add it to the waiting time of other trucks.

Modifications to do:

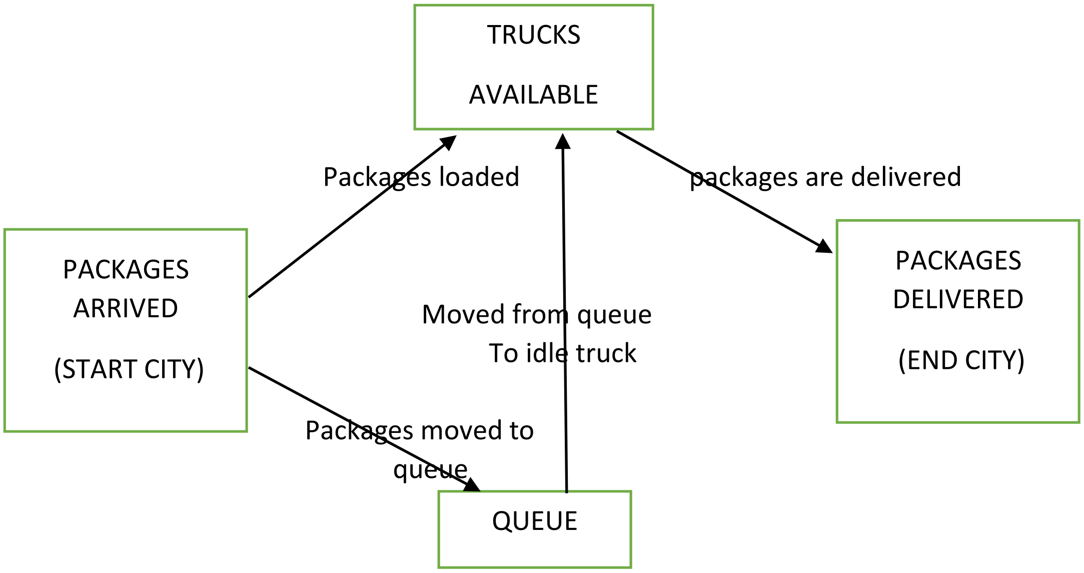
- Both the truck's and the packages status is updated to "delivered".
- Time solts of the truck is updated appropriately
- Package delivery time is included as the current time.
- There is also a package- delivery- counter maintained which is incremented as and when the packages are delivered, to maintain the count of packages delivered.

**Truck Ready to Pick Up:**

If a truck delivered all the packages it is carrying then it is made ready to pick up the next arriving package by updating its slot values. So that the values updated by the truck after picking up the next package must be absolute.

Modifications to do:

- Its status is made "idle".
- Available space is made to the same as before it picked up packages
- It has no destination city and no packages carrying



**DESIGN OF PACKAGE DELIVERY SYSTEM**

Below is the design of the facts used to simulate the design mentioned

Given map is prepared in the form of facts (city) with slots defining city name and the cities to which it is connected to along with the distance between them.

(city)

- City_name- name of the city
- Connected_cities- list of cities to which the city is directly connected to along with the distance between them

From this data collected shortest path among all possible cities is calculated by using four rules defined in rules file along with other rules

(shortest_path)

- City1- start city from which distance must be calculated
- City2- end city to which distance must be calculated
- Time- min time to travel from start city to end city
- Via- list of cities in between start city and end city shortest path

Truck details are specified as facts (truck) with slots

- truck_no- defines truck number
- current_city- trucks current city
- multislot destination_city- all those destinations truck must visit to deliver packages it is carrying
- current_action- defined to be “idle”, ”going to pick a package”, ”in route to deliver a package”, “delivered package ”. to best fit all the scenarios while delivering a pacakge
- waiting_time- truck idle time without a pick to pick up or deliver
- busy_time- time spent for travelling, while picking up and delivering the package
- reached_time- time to reach the next destination, used to update the clock
- Available_space- space available in the truck after picking up the package
- space_occupied- space occupied by the packages which truck currently carrying
- packages_carrying- Total number of packages a truck delivered in total time
- tot_space_occupied- total amount of truck’s space utilized by the packages it delivered. It may suggest the admin whether to place a truck with more capacity among particular routes so as to reduce the waiting time of the package
- non_del_time- travel time spent by truck to pick up a truck
- del_time- time spent by truck to deliver the package after it picks it up
- numofpack- list of packages a truck is carrying at present time

Package details are specified as facts (package) with slots

- package_no- defines the package number
- depart_city- city to which the packet is arrived
- deliver_city- city to which the packet must be delivered
- size_of_package- package size, given in the data set
- arrival_time- time at which the packet is arrived
- exp_delivery_time- time at which the packaet is expected to be delivered
- ppick_up_time- time at which thepacket is picked
- pdelivery_time- time at which the packet is delivered
- wait_time- time for which the packet waited for it to pick up(also in queue)
- status- defined it to be “arrived”, “waiting to pick up”, “in-transit”, “delivered”

**Output Report:**

Output is generated in tabular format which has easy readability. Truck report consists of details about every truck participated and non-participated in the simulation. The results are generated in the ascending order of the truck number. Header for the truck report is formatted in the rule which triggers only after all trucks has delivered all the packages carrying and package- delivered counter is equal to the packages arrived. Below is the description of the truck report columns

- Truck Report
  - Truck_no- truck number in ascending order
  - Wait-time- total amount of time for which truck is idle
  - Total-busy-time- total time spent by truck to pick up and deliver the package
  - %busy- time
  - Total-packages-carried- total number of packages a truck delivered
  - Avg-occupancy- Average space occupied by packages a truck has delivered
  - Non-delivery travel time- time spent by truck to pick up packages
  - %non-delivery-time
  - Delivery-travel-time- time spent by truck in delivering packages after picking them up
  - %delivery-time

Similarly package report describes about the packages details in ascending order of the package number. It shows what are all the packages delayed their delivery, by how much time, time at which they got to be delivered etc. below are the columns specified in Package report.

- Package Report
  - Package number- package number in ascending order
  - Total-wait-time- total wait time of package after its arrival
  - Pick-up-time- time at which package is picked by a truck
  - Delivery-time- time at which package is delivered by the truck
  - On-time delivery-status- states whether package is delivered on time. True or false
  - Delivered delayed by- time by which a package is delayed in its delivery

A package average report is also used generated to display the average time of all packages like average lateness, packages delivered on time etc.

- Package Average Report
  - Avg-wait-time- Average wait time of all packages
  - Number of packages delivered on time- count of packages delivered on time
  - Number of packages delayed- count of packages delayed
  - Avg lateness of late packages- average of the delayed time of delayed packages
  - Avg lateness of all packages- average of the delayed time of all packages

## Specific Changes Made:

### Functionality changes:

- Shortest path calculation is automated. The fact created is shortest_path and it includes city1 the start city, vity2 the destination city, time taken to reach in the shortest path and via includes the path considered to reach destination. The slot via city helps truck to travel city by city and pick any new packages arrived to via city. This reduces the packet waiting time which allows it to be delivered in the time expected.
- A queue concept is introduced to handle the situation when there is a package arrived and no truck idle with capacity greater than package size to pick it up. Then the package is put into the queue. The time spent by the package in the queue waiting for it to pick up is added to the package waiting time. After any truck becomes idle then package from the queue in first come first serve order is assigned to that truck.
- If any two packages waiting in the queue have same arrival time then the package which is expected to be delivered first is assigned to the truck.
- If any two trucks becomes idle at the same time then the truck which can reach the package early is allowed to pick up. The status of the package waiting in the queue will be “waiting in queue”.
- All the packages entering the queue are added to the end of the queue whereas packages assigned to trucks are picked up from the front of the queue, thus implementing queue property first come first serve.
- Truck can pick up more than one package which has arrived on its way to destination city if it has enough space to pick them. The destination city slot is adjusted as per the distance between the newly picked up packages. If this can be delivered first the truck behaves accordingly and delivers packages one after the other.
- Situation when two packages arrived at same time to different places then the trucks nearest to the packages arrival city will pick them up accordingly.
- Trucks are made not to start at a single place. Indeed they are spread among the cities. Also a truck need not come to the current city again after delivering a package for it to pick up next package. It stays at the same place where it delivered the package carrying and waits for the next package to be arrived.
- The success condition for all packages delivered successfully checked by incrementing the counter variable after delivering the packages. This counter value is checked by hardcoding the value is the rule in part1 but in part2 there are two global variables introduced in the dataset to check the packages counter whether all packages has been delivered or not. By this way the code is made more efficient.

**Changes in Facts:**

- Added a new fact shortest_path to handle shortest distances calculated among cities.
- Slot destination city of the truck is made multi slot to hold multiple destination cities it has travel to deliver the packages carrying.
- Slot tot_space_occupied is added to the truck to maintain the total space occupied by packages it delivered which is used in report to calculated average % occupancy of the truck
- Multi Slot numofpack is added to truck to maintain the list of multiple packages it is carrying.
- Two global variables are used in the data set to maintain the number of packages arrived and number of trucks present to deliver them.
- Output is displayed in tabular form, few values are added to the truck's report which will be useful in determining the efficiency of the simulation like Average % occupancy, which may need to know whether to include a truck with large capacity in particular areas; Non-delivery time and delivery time of the truck along with its percentage figures.
  Output generated for the data sets provided are attached here for reference

  output_ds1 output_ds2 output_ds3

## Rational for the Changes Made:

- Reason behind the introduced heuristics of part2 is mainly aimed to reduce the waiting time of the packages and to keep trucks busy in delivering as many packages as possible. Below mentioned are in-detail rational for the changes mentioned above.

- Shortest distance is automated as this increases the scope of the map used to deliver packages. If in future the same module is to be implemented for another state, or if any more number of cities are included in same city then just by adding them to the facts that described map would be sufficient, it automatically calculated the shortest distance and route between those newly adding cities without effecting the existing one. Also, each and every combination of the cities is considered in this calculation. Hence probability of making any mistakes during this calculation is very low, where as human calculated distances may sometimes be wrong as there is high possibility for human to make mistakes.

- By considering the possibility that a truck can carry as many number of packages that fits into space available the change for a truck to pick up other packages on its way is been implemented. This also reduces the waiting time of the package allowing to increase the statistics of the on-time delivery packages. Max utilization of the truck can be seen. By this way there will be less number of packages waiting in the queue for a truck to come a pick them.

- Implemented queue concept to follow the proper flow of the delivery system. There should be no random method to pick up and deliver the packages. Queue is implemented to maintain the systematic pickups and deliveries. Queue allows the truck to choose correct package to deliver.

- Current city of the trucks is not same, they are distributed among the cities present in the map. Rational for this change is to improve the package waiting time and truck non – delivery time.

- To pick up the packages arrived on same time nearest trucks are assigned to pick them up them accordingly, this is made working in the part2 as part1 doesn't handle the situation. In part one the second package arrived at the same time enter into queue leading to infinite loop. Whereas part2 is made working with queue.

## Improvements That Have Resulted From These Changes:

**Good Improvements:**

- After making the above mentioned changes to part1, the overall average waiting time of packages has been reduced. Number of packages delivered late has been come down. The system is working with a well-defined architecture. The efficiency of the design is improved on a larger scale.
- As the trucks are spread across the cities and the nearest truck is assigned to pick up the package whenever a package arrived the truck decides to go pick it up, then no other truck need to start. Hence the package waiting time for a truck to pick up reduces.
- By storing the package number of the package the truck is carrying it makes sure to deliver the same package as in part1 in the rule truck-deliver-package it is checking only the truck destination city and package deliver city to be the same there may be a situation where the condition given is satisfied but the package may be different, this has been corrected in part2 and now the truck only delivers the package it is carrying.
- Introduction of queue system made packages to be delivered in a particular order. This will be helpful when more number of packages are introduced for the simulation.
- Increase in number of trucks also reduces the waiting time of the packages, since more number of trucks will be spread across the cities where they are ready to pick up one immediately after it arrives, or else they should be waiting for trucks to become idle.
- Now any number of packages can be delivered by using this program and design

**Bad Improvements:**

The heuristic of delivering the truck city by city is not following the complete path from arrival city to deliver city of the package. It only visits to the first via city and from there it delivers directly to the destination city of the package. So, if any package arrives to the first city in the via list then it picks up and delivers according to the destination of the both packages whichever is on the way. The situation of visiting the package's deliver-city, city by city from start city is hard to implement without using functions and loops. The design needs to check the size if the slot via city after calculating shortest path and then the rule must behave according to the size of the via slot to know how many cities needs to be visited to reach the deliver city. A loop must be implemented to visit the city in between path and check if a package has arrived that particular city, if so then pick it up too if space available is sufficient or else assign it to the next nearest truck.

## Different Idea:

If there is another chance to implement this program, I would have made my facts simple by separating one from other. Currently, the fact shortest path includes four slots including the path to follow to destination. This makes the rules implementation complex while a truck needs to pick up packages which arrived on its way. The movement of truck in between such existing cities is being hard to check. So representing data in a simple way makes like easier.

Also I shall try to update the clock step by step and check if there is any event to take place at that time. Or else just increment the time by 1unit. Following this way it is easy to check each and every scenario to deliver the package in the earliest possible way to its destination.

A package can be transferred among the trucks, if by doing so it may reach the destination early. At every clock update it should be checked whether any two trucks came across each other, if so then check the destination of the packages they are carrying. If other truck is travelling towards the destination city of the package a truck is carrying then simply transfer that package to the other truck. In this way the probability of the package reaching its destination early will increase. As our goal is reducing the number of packages delayed in their delivery.  This would lead us to ideal systems which can used in real time applications.

It would be much useful to store the truck statistics according to their location of travelling to know higher number packages has arrived in which city and which is the most visiting city of deliver package. By this data it is easy to know where to place the trucks initially and also whether the trucks capacity has to be increased before placing them in a city.