Senior Design 2009

Scheduling Agency Pick-ups at the North Texas Food Bank

CLOSE THE GAP

50 Million Meals By 2011.

Presented by:

Allison Griffin
Pragya Lohani
Megan Walker

SMU BOBBY B. LYLE SCHOOL OF ENGINEERING

Table of Contents

Project Summary

The North Texas Food Bank (NTFB) is a non-profit hunger relief organization that distributes donated, purchased, and prepared foods through a network of feeding programs in 13 North Texas counties. From these counties nearly 700,000 individuals live below the federal poverty level. The North Texas Food Bank's mission is to passionately pursue a hunger-free community; more specifically their "Close-the-Gap" initiative is to make 50 million meals available annually.[1] This project seeks to solve one problem at the food bank that will hopefully help the organization achieve this goal more efficiently and quickly.

During our initial meeting with the Director of Operations, Mr. Sean Gray, we toured the warehouse to understand the daily operations at the food bank. Our objective was to locate the bottleneck and the steps that were creating inefficiency in the process. From the different problems that arose such as facility design and layout, inventory management, processing of orders, food distribution, and forecasting, we realized that streamlining the food distribution method for the agencies that pick-up from the warehouse would be the most beneficial for the NTFB to reach their target goal.

We spent time at the food bank collecting time data, analyzing distribution data, and observing the current process for the agency pick-up. There were several inefficiencies in the process that led us to believe that if the food bank and agencies had a schedule to work by every week, then time could be used more efficiently for both parties.

The model that we developed allows the NTFB to generate a schedule for their agencies in a given week. Variations can be made to the model to accommodate NTFB preferences, agency preferences and other situational constraints. With this schedule and the more stream-lined process we offer in this report, the food bank will be in a better position to cater to their agencies and reach their target goal.

[1] A non-profit hunger relief organization distributing donated food - North Texas Food Bank. 05 May 2009 <http://www.ntfb.org>.

Background and Description of the Problem Situation

The NTFB has approximately 260 agencies pick-up their food orders at the warehouse weekly. Currently, the NTFB distribution center is open 3 days a week (Tuesday, Wednesday and Thursday) from 7 am to 12pm for pick-up. Gray informed us that these days and times are not a set restriction to the pick-up process, but the staff involved need to be considered if the times are to be changed. The procedure begins when an agency places an order online before Sunday at 11:59pm based on current stock for a given week. The NTFB approves the order and agencies are expected to collect their order on one of the three given days, when the distribution center is open.

As the agencies arrive, orders are loaded on a first come, first serve basis. So if an agency with 20,000 lbs of order arrives at 7 a.m. in the morning then NTFB begins to load their order first even though many other agencies have smaller orders. The loaders work on several orders at once, which would make sense if they were using more than one dock or forklift. Instead, parts of various agencies' orders are being prepared to go out of one dock, using one forklift. This causes a bit of chaos on days where many agencies show up at one time, and really eliminates the first come, first serve process. The loaders frantically put together and get out parts of all of the orders, instead of trying to complete one agency's full order. Therefore, if the fifth agency to arrive only has one load of pallets, they may finish before the first agency that had three loads. The agencies do not see this as fair and the food bank is not really complying with their promised first come, first serve process. This issue creates problems such as: a long waiting line, a full parking lot, and chaos all around. We believe this type of order-processing and loading is not completely stream-lined and is creating inefficiency in the system.

While loading the orders that were placed online, which are prepared before the agencies arrive, NTFB loaders also prepare pallets of frozen and produce items that are ordered upon arrival. Because all pieces of an agency's order may not be ready, this is often the reason that an agency is not fully serviced before moving to another. But, if an agency had a schedule time of arrival, then the NTFB could prepare the pallets in advance, increasing the efficiency of the whole process.

Besides these main processing problems, the NTFB currently uses either one or two docks for loading despite having five docks available. The NTFB also has more than one forklift but uses only one in the loading process. These considerations would create the need for more workers to run the forklift, but we believe the new process will eliminate the need for so many workers inside the warehouse preparing orders. After analyzing the situation at the NTFB, we decided to come up with a model that would restructure order-processing and loading.

Analysis of the Situation

Our general approach to the problem started by mapping out the current process at the NTFB by using flow charts. Afterwards, we rigorously timed and analyzed each process with time cards and actual observations. An example of a flow chart that maps the entire process is shown below:

Figure 1: Current Process Layout at NTFB

We considered each step in the process and discussed with food bank employees options that could increase productivity and efficiency at each stage. We analyzed bottlenecks and thought about solutions to the problems we noticed. The actual observations provided us with important data such as the average loading rate, average time for pick-and-weigh and various other data crucial to the model. This time data also showed us the bottlenecks and places where the most time is lost during the process.

We quickly realized that the NTFB and agencies would benefit from a schedule that provided the staff and clients with a more clear vision of how a week will run and when an order will be ready. The model could provide an in-depth analysis of the effects of opening up more than one dock for loading, adding another distribution day, or even both. The model could run with different variations according to a particular situation the NTFB may have or want to create for a given week. It can also incorporate preferences of the agencies as to which day they have staff available and would prefer to pick-up at the NTFB.

We hope that the use of this model will reduce the wait-time for the agencies and will provide a schedule that the NTFB staff can use to be more organized and prepared for the customers. This creates a whole new process for distribution of the food that will be more efficient (see Figure 2 below). When an agency arrives for their pick-up time they will immediately go to the pick-and-weigh station while the loaders begin putting together their frozen items and produce pallets. These additions to the order will be added to the other pallets waiting to be loaded at the assigned dock. The agency will have their vehicle pulled directly up to the dock now because they are assigned to that dock at that time and the load can be put directly into their vehicle, avoiding any excess driving around the parking lot or returning of emptied pallets by the loader.

By creating efficiency in all of the stages of the process, we believe that overall this model will restructure the order-processing and loading at the NTFB. The schedule is created by the generation of patterns that include the timeslots available for a given amount of days and docks. The input is the number of timeslots an agency would need to load the amount they ordered online plus the amount of their pick-and-weigh and additional produce from a forecast by agency size. Once feasible patterns have been created, the patterns are compared to see which ones are compatible, meaning that they could be used during in the same schedule. Patterns that use the same time periods could not be used on the same schedule for two different agencies, so the model notes those as incompatible. Then, the model finds those patterns that each agency can be assigned to based on the amount of timeslots an agency requires, and also denotes all of the agencies that can be assigned to a given pattern.

At this point the model has all of the patterns that are possible for each agency. To develop the actual schedule the model uses constraints that require that only one agency be assigned to a given timeslot pattern, including a day and a dock, and that an agency can only be given one feasible pattern. The objective function gives costs to the patterns, docks and days so the schedule is filled in the most optimal order. For our analysis, we placed the most emphasis on early timeslots, and then filling the days, with the docks as the last concern. We also included a preference on the day an agent would like to pick up which could be given as an input also. Although this request is not forced as a constraint, the preferences allow it to be chosen first.

A small example of the model could be three agencies that have a requirement of 1 timeslot, 3 timeslots and 10 timeslots, and preferences of Day 1, Day 3, and Day 1, respectively. The model is given the situation of 3 days, just one dock, and up to 20 timeslots per day per dock. Of course in this model, there are more than enough timeslots for the small demand these agencies have, but more docks could be opened for a higher demand of timeslots. The model would first create the patterns described in the more technical description later in this report, and assign the agencies to possible compatible patterns. In a model that placed more preference on the days, Agencies 1 and 3 would be given the first 11 timeslots in Day 1 and Agency 2 would be given the first 3 timeslots in Day 3 because of its preference. If the model placed more emphasis on the docks, the solution would spread out the work by placing Agency 1 in the first timeslot of Day 1, Agency 2 still with its preference on Day 2, and Agency 3 in the first 10 timeslots of Day 3. There are numerous ways to change the solution based on various costs on each factor, but for this analysis we tried to match the situation and preferences the food bank expressed to us throughout our time there.

Technical Description of the Model

The first part of the model is simply a generation of the patterns that create binary data of potential time periods to assign the agencies to. There is a set of patterns, set GROUP, for each number of timeslots increasing by one timeslot for each new group, with the last group having the largest number of timeslots required by any agency, called MaxTime. The incompatibility function runs a loop through all of the columns of patterns for each row, P[r,c_i], where P is the pattern, r is the row and c_i is the column i. It looks for two columns in a specific row that are both one and sets them as incompatible so that later in the model they will not be used at the same time. For example, if P[r,c1] + P[r,c2]=2, then c1 and c2 are set to incompatible. [2]

At this point, the model denotes patterns that an agent, a, could possibly be assigned, PA[a], based on the amount of time periods an agency needs to fill their order, which is an input parameter, param time. The model also denotes agents that can be assigned to a given pattern, AP[c], where c is the pattern. This information is simply from the incompatibility constraints and patterns that were created.

There are three constraints listed below that make up the rest of the model. Two binary variables, x{a in Agents, PA[a], Docks, Days} and y{c,Docks,Days}, are created to make the assignments in the constraints. Constraint one requires that an agency only be assigned to one pattern, one dock, and one day and sets the values for the binary variable x. The second constraint sets the binary variable y for a given pattern, dock and day equal to the x variable for the corresponding subscripts. Finally, constraint three ensures that incompatible patterns, ppi, as found in the incompatibility constraint, are not both assigned to a given y.

Constraint 1:
subject to C1 {a in Agents}: sum {p in PA[a], d in Docks, Dy in Days } x[a,p,d,Dy] = 1;

Constraint 2:
subject to C2 {p in C, d in Docks, Dy in Days}: sum{a in AP[p]} x[a,p,d,Dy] =
y[p,d,Dy];

Constraint 3:
subject to C3 {(pp1,pp2) in INC, d in Docks, Dy in Days}: y[pp1,d,Dy] + y[pp2,d,Dy]
<= 1;

The objective function, shown below, minimizes the costs assigned to each aspect of the assignment process: the patterns, docks, and days. The multiplier, A, is the cost given to the patterns, p, and we assigned this as the highest cost in order to place the most emphasis on

[2] Kennington, Jeffrey. "Senior Design."

assigning an early time period. This cost forces that early time slots be filled first, and in order, so that a day is filled sequentially. The cost parameters for the docks and days, costDocks[d] and costDays[dy], can be adjusted to place more emphasis on one or the other. The effects of changing costs are discussed later in the analysis.

The agency day preference is also considered by multiplying the cost parameter for the days times a cost parameter for the agency day preferences, costADy[a,Dy], which will assign an agency to the day they request if it is feasible and skip the normal filling of the days for that agency.

Objective Function:
minimize COST: sum{a in Agents, p in PA[a], d in Docks, Dy in Days}
(A*p+B*costDocks[d]+costDays[Dy]*costADy[a,Dy])*x[a,p,d,Dy];

The sample used in analysis was created based on a week of historical data from the food bank's Navisions Financials database to create realistic, hypothetical values. We pulled the amount of pounds each of 60 agencies picked up during a week at the food bank. The amount of timeslots required for each agency, the input of the time parameter found in the sample data file in Appendix A, was calculated based on this poundage and the loading rate we calculated from our time collection data. The assumption in our model is that 1250 pounds can be loaded per timeslot, which represents 15 minutes. Because the loading rate is not a definite calculation and could vary based on different time data, the amount of pounds that can be loaded in a given timeslot could also vary. The number of variables in the model is based on the number of agencies, days, and docks used, but for the purpose of our analysis we used just 60 agencies, and any range from 1 day and 1 dock to 4 days and 4 docks. The model was run using AMPL and a full version can be found in Appendix B.

Analysis and Managerial Interpretation

The flexibility of our model allows for multiple variations to suit the NTFB's weekly demands. However, only four variations of the model will be used to provide a general overview of the model and its usage. Agency preferences for arrival times are included for all of the following models and remain constant. Since we did not have actual preferences, the ones currently being used in the model are fictional, but real ones could be collected at the time an agency places an order online. The outputs of these four variations will be used to generate recommendations for managerial purposes.

- Model 1 – Model 1 is a demonstration of the Food Bank's current methods of distribution. This current method has three days of distribution with five hours of operation each day. They presently have only one dock and one forklift open for loading.

Figure 2 below is the output of this model, showing that it is actually infeasible based on the loading rate we calculated. This would explain the rushed nature of a full day at the warehouse, customers leaving before their orders are loaded, dissatisfied customers, and stressed workers.

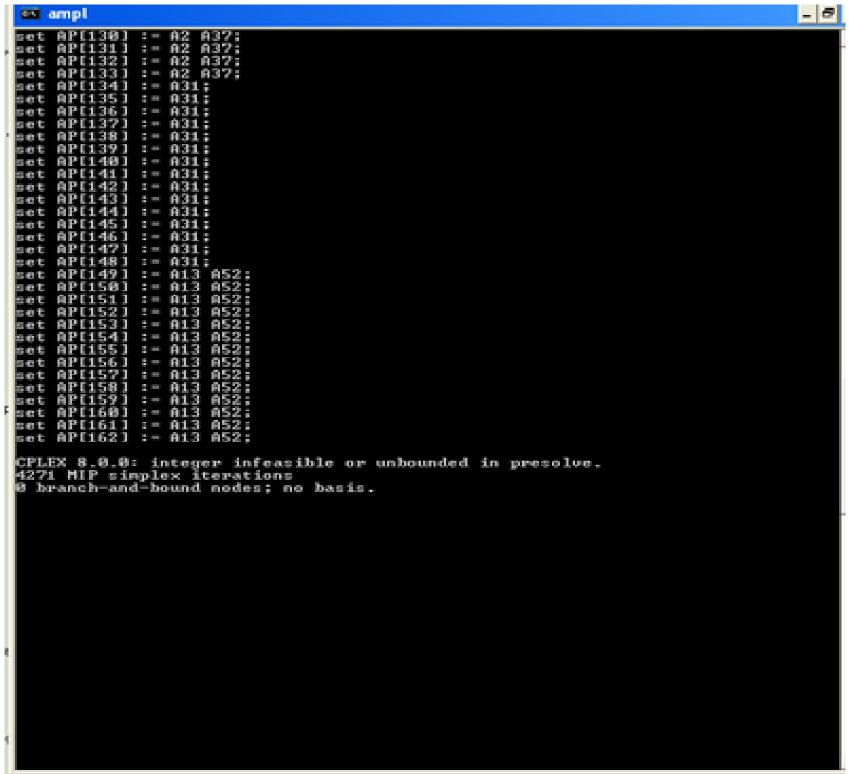

Figure 2

- Model 2 – This next variation has the NTFB open for 3 days with 2 docks available for loading. More emphasis is being placed on the docks, so the model fills up early timeslots on each dock first and creates a more evenly distributed workload for the week. Also, the food bank is able to service all of their agencies by 12:30 on each day. Figure 3 displays this model's output.

Time Slots	Day 1 Dock 1	Day 1 Dock 2	Day 2 Dock 1	Day 2 Dock 2	Day 3 Dock 1	Day 3 Dock 2
7:15	A 54	A 47	A 39	A 35	A 6	A 43
7:30	A 29	A 27	A 25	A 10	A 28	A 26
7:45	A 24	A 57	A44	A 59	A 23	A 33
8:00	A 22	A 17	A 60	A 15	A 41	A 48
8:15	A 40	A 7	A 9	A 50	A 38	A 16
8:30			A5		A 18	A 53
8:45	A 12	A 42	A 14	A 19	A 21	A 46
9:00					A 58	A 8
9:15	A2		A 34	A4	A3	A 51
9:30		A 32			A1	
9:45			A 30		A 56	A 55
10:00				A 49		
10:15					A 11	
10:30		A 52	A 45			A 13
10:45				A 31		
11:00	A 37				A 36	
11:15			A 20			
11:30						
11:45						
12:00						
12:15						
12:30						

Figure 3

- Model 3 – This model has three days of distribution with two docks opened. In this variation, NTFB has placed stronger emphasis on the days, meaning it fills the days sequentially before filling the docks as in Model 2. This may be because they can or want to schedule more employees for a specific day, or because they know they will have a lack of labor on certain days. After running the model, the following schedule was given. As you can see, because of the NTFB preference the food bank is open until 2:30pm on Day 1, yet it is closed by 11:00am on Day 2 and by 12:00pm on Day 3.

Time Slots	Day 1 Dock 1	Day 1 Dock 2	Day 2 Dock 1	Day 2 Dock 2	Day 3 Dock 1	Day 3 Dock 2
7:15	A3	A 24	A 25	A 15	A1	A 46
7:30	A 44	A 59	A 60	A 10	A 48	A 26
7:45	A 38	A 53	A 40	A 50	A 21	A6
8:00	A 54	A8			A 16	A 41
8:15	A 23	A 17	A 34	A19	A 56	A 51
8:30	A9	A 22				
8:45	A 39	A 29	A 55	A 30	A 11	A 36
9:00	A 28	A 58				
9:15	A 35	A 57				
9:30	A 18	A 47	A 45	A4	A 31	
9:45	A 27	A 43				
10:00	A 33	A5				A 13
10:15	A14	A 12	A 49	A 20		
10:30						
10:45	A7	A 42				
11:00						
11:15	A2					
11:30		A 32				
11:45						
12:00						
12:15						
12:30		A 52				
12:45						
1:00	A 37					
1:15						
1:30						
1:45						
2:00						
2:15						
2:30						

Figure 4

- Model 4 – This final variation looks at a situation where the food bank is open for 3 days, has 4 available docks, and a preference for distribution on Day 1. Opening more docks allows the food bank to decrease their hours of operations, sometimes drastically. Day 1, which was given preference, is done with distributions by noon. On Day 2 and 3 they are closed as early as 9 am. Therefore, this model gives them the flexibility to change their hours of operations or the number of days they are open.

Time Slots	Dy 1 D 1	Dy 1 D 2	Dy 1 D 3	Dy 1 D 4	Dy 2 D 1	Dy 2 D 2	Dy 2 D 3	Dy 2 D 4	Dy 3 D 1	Dy 3 D 2	Dy 3 D 3	Dy 3 D 4
7:15	A8	A33	A27	A17	A 14	A 45	A20	A49	A41	A6	A16	A26
7:30	A54	A39	A9	A48					A21	A51	A46	A 56
7:45	A23	A25	A 57	A 60					A1		A36	
8:00	A24	A38	A5	A44		A55			A13	A11		A31
8:15	A30	A35	A 47	A28								
8:30	A53	A 58	A22	A 59								
8:45	A10	A40	A49	A18		A15						
9:00	A3			A12								
9:15	A7	A19	A34									
9:30				A 42								
9:45	A4	A52	A37									
10:00				A32								
10:15												
10:30	A2											
10:45												
11:00												
11:15												
11:30												
11:45												
12:00												

Figure 5

The model and its variations provide the food bank with schedules that enable them to operate efficiently so that they are able to service as many agencies as they have the potential to. By making few and minor changes in the model, they have the ability to increase or decrease the number of days of distribution and docks to cater to their agencies' demands accordingly. In the final model, we saw that the distributions were done by 9:00 am by opening up just 2 more docks. This means that the NTFB could make such changes and by doing so have more time to service a higher number of agencies.

Conclusions and Recommendations

After analysis of the process and our scheduling model, we highly recommend the implementation of a weekly pick-up schedule. With a schedule, several time efficiency problems will be eliminated and a new process can be implemented. Find a flow chart of the new process in Figure 6 below, as compared to the current process shown above in Figure 1. Agencies will arrive at a given time with their online order already processed waiting at their assigned dock. Upon arrival, workers will be notified to start processing frozen and produce items immediately while the agency shops in the pick-n-weigh area. A variation to this would be that the frozen items be prepared before the agency actually arrives, but the food bank still expressed concern about these items sitting out in case an agency does not show. After shopping, the agency will move their car to their assigned dock while the workers add the pick-n-weigh and frozen pallets to the already processed order. The agency's entire order will then be placed directly in their car from the lift, eliminating the time lost as the forklift moves around the parking lot in the original process.

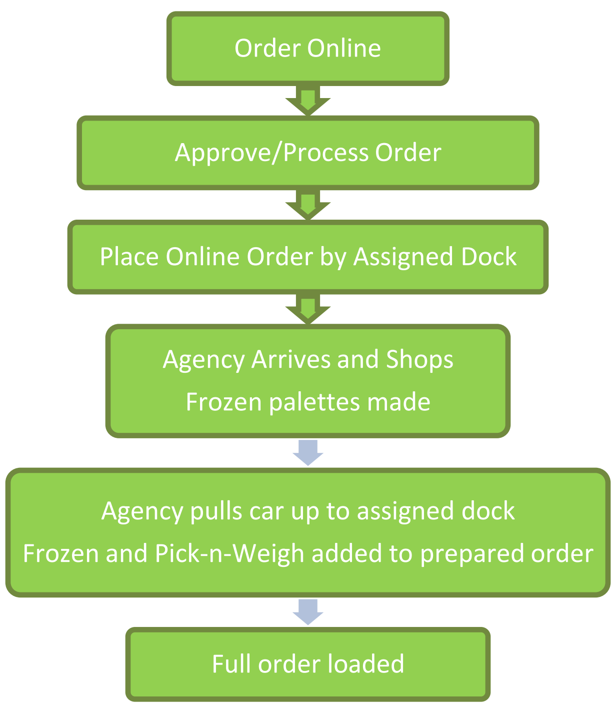

Figure 6

In some of our last days of time data collection, we experienced an extremely busy day and got a lot of feedback from the agency representatives. This observation and feedback helped us gain insight into exactly how much our model can improve the day-to-day operations of the NTFB. On this particular day, a worker was missing on the inside of the warehouse that usually helps to prepare orders, and the forklift driver was less experienced than the usual driver. These two problems created a back-up that could have been lessened if our process and schedule recommendations were implemented. First, if our process was implemented the driver would not have to maneuver the forklift all over the parking lot as the agency vehicle would be pulled up to their assigned dock. But more importantly, the problem that we noticed on this day was that the loaders kept starting agency orders as new agencies arrived without finishing orders they had previously began. This led to a very long line of angry agencies in the parking lot, filling the waiting room and all along the loading dock area. If our schedule had been implemented there would not be this back up of agencies waiting. But also if our new process was used, an agency would be fully serviced at its assigned dock before moving onto another one. More than one agency could be serviced at a time on a day like this with our model, but it would be done by opening more docks. The agency representatives we talked to seemed more than willing to adhere to a given pick-up time, and were actually excited about the prospect of a schedule.

We were also able to witness the implementation of our recommendation to open another dock. Our entire process was not implemented but with another dock open, which includes using another forklift, the day definitely ran more smoothly and the waiting room remained fairly empty.

There are a few effects that need to be considered in the future for our model. We did not include worker breaks, which could easily be implemented in the model. A fake agency could be entered in the input that actually assigns break time for the workers instead of an agency coming at that time period. Also, the loading rate that we calculated is relative to the days we got the data. The time data collected to calculate this factor depends on the workers, the amount of agencies at the food bank which can create a sense of urgency, and the size of the agencies that were there on that particular day. We also recognize that loading rate could also be associated with the number of pallets being loading instead of the actual poundage of the load which is the factor we considered. The loading rate also differs for a car versus a truck that is pulled directly up to the loading dock, which was not considered. With this idea in mind, we also thought about forcing the model to assign all agencies over a certain large number of pounds to a given dock. This would create a dock that only services large agencies and another way to create a more stream-lined process.

Generating schedules is crucial for the food bank to implement our recommended process. Whether or not the model we have created is used, a new process and schedule is necessary for the successful growth of the food bank in order to meet their goal of 50 million meals per year by 2011.

Appendix A

Sample Data File

```
set Agents := A1 A2 A3 A4 A5;
param TimePeriods := 14;
param Time:=
A1 10
A2 3
A3 5
A4 8
A5 1;
set Days := Dy1 Dy2 Dy3;
set Docks := D1 D2 D3 D4;
param costDocks := D1 1 D2 2 D3 3 D4 4;
param costDays  := Dy1 1000 Dy2 2000 Dy3 3000;
param costADy   :=

A1 Dy1 3
A1 Dy2 2
A1 Dy3 1
A2 Dy1 1
A2 Dy2 3
A2 Dy3 2
A3 Dy1 2
A3 Dy2 3
A3 Dy3 1
A4 Dy1 1
A4 Dy2 1
A4 Dy3 1
A5 Dy1 1
A5 Dy2 1
A5 Dy3 1
;
```

Appendix B

Sample Model

```
set Agents;
param TimePeriods;
param Time{Agents};
set   Docks;
set Days;
param costDocks{Docks};
param costDays {Days};
param costADy{Agents,Days};

data datafile.txt;
display Agents, TimePeriods, Time;
display Docks;
display Days;
display costDocks;
display costDays;
display costADy;


param MaxTime;
let MaxTime := 0;
for {a in Agents} {
  if Time[a] > MaxTime then let MaxTime := Time[a]; } display MaxTime;

param Columns;
param Temp;
let Temp := TimePeriods-MaxTime;
let Columns := (TimePeriods*(TimePeriods+1))/2 - (Temp*(Temp+1))/2; set C; let C :=
1..Columns; set R; let R := 1..TimePeriods;
param P{R,C} default 0;     # P gives the patterns for schedules
display R, C;
display P;

# Construct Patterns (schedules)
set GROUP;
let GROUP := 1..MaxTime;
param col;
set NG;
param begin{GROUP};
set I;
let col := 1;
```

```
for {group in GROUP} {
  let NG := 1..TimePeriods+1-group;

#   NG gives the number of patterns in each group
  let begin[group] := col;
  let I := 1..group;
  #   I gives the number of 1s in the current col
  for {ng in NG} {
    for {i in I} {
      let P[ng+i-1,col] := 1;
    }
    let col := col + 1;
  }
  for {r in R} {
    printf"\n";
    for {c in C} printf"%2d",P[r,c];
  }
  printf"\n\nbegin[%2d] = %3d \n",group, begin[group]; }

# If two paterns (schedules) use the same time period, # then they are incompatable set INC within {C,C}; let INC := {}; set INC within {C,C}; let INC := {}; for {c1 in C} {
  for {c2 in C: c2 > c1} {
    for {r in R} {
      let Temp := P[r,c1] + P[r,c2];
      if Temp == 2 then {
        let INC := INC union {(c1,c2)};
        break;
      }
    }
  }
  display c1;
}

display INC;

set PA{Agents};   # PA[a] denotes the patterns that agent a can be assigned
param t;  param p1;  param p2;
for {a in Agents} {
  let t := Time[a];

let p1 := begin[t];
  if t < MaxTime then
    let p2 := begin[t+1] - 1;
  else
```

```
 let p2 := card(C);
 let PA[a] := p1..p2;
}
display PA;

set AP{C};  # AP[c] denotes the agents that can be assigned to pattern p for {c in C} let AP[c] :=
{}; for {a in Agents} for {c in C} let AP[c] := {}; for {a in Agents} for {c in PA[a]}
   let AP[c] := AP[c] union {a};
display AP;


var x{a in Agents, PA[a], Docks, Days} binary; var y{C,Docks, Days} binary;

#Constraints
subject to C1 {a in Agents}: sum {p in PA[a], d in Docks, Dy in Days } x[a,p,d,Dy] = 1;

subject to C2 {p in C, d in Docks, Dy in Days}: sum{a in AP[p]} x[a,p,d,Dy] = y[p,d,Dy];

subject to C3 {(pp1,pp2) in INC, d in Docks, Dy in Days}: y[pp1,d,Dy] + y[pp2,d,Dy] <= 1;

#objective function
minimize COST: sum{a in Agents, p in PA[a], d in Docks, Dy in Days}
 (100*p+10*costDocks[d]+costDays[Dy]*costADy[a,Dy])*x[a,p,d,Dy];

solve;

#Output
printf"\n\n";
for {a in Agents} {


 for {p in PA[a]} {
   for {d in Docks} {
          for {Dy in Days} {
     if x[a,p,d,Dy] == 1 then {
       printf"Agent %3s is assigned time slot(s) on Dock %s on Day %s  ",a,d,Dy;
       for {r in R} {
         if P[r,p] == 1 then printf" %3d ",r;
            }
           }
     }
   }
 }
 printf"\n\n"; }
```