Project Report
(under DAAD Wise Scholarship)

University:Technical University of Munich,
(School of Management),
Munich,
Germany

Department:Operations Management and technology

Professor: Dr. Renzo Akkerman

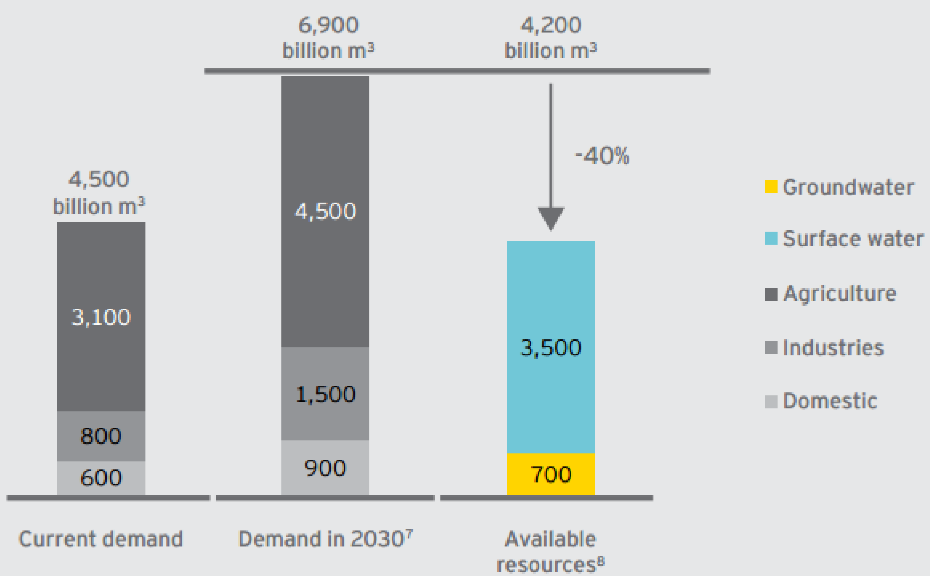
Supervisor: Pulluru Sai Jishna

By,
S.Karthik Sundaram,
Production Engineering,
National Institute of Technology ,India

Introduction and abstract:

The main aim of this approach is to address and redress the situation of water scarcity using water reuse .

Figure 1: Water demand and supply 2030 projections



Source: 2030 Water Resource Group and International Food Policy Research Institute (IFPRI)

As can be seen in the above case water demand is seen to increase according to the forecasts whereas the supply is remaining a constant. Hence in order to sustain in the long run economically and as a part of the environmental conscious motive, all the companies should employ and concentrate more on waste water management and also improve their processes to reduce the initial water consumption. This can be done by a lot of data analysis techniques and mathematical modelling to optimise the water usage. For doing this the first steps should be to analyse the current scenario by representing the whole company in a mathematical model and collecting data including the time logs and amount of water consumed with respect to its contaminant concentration.

Waste –1) whey production
2) cheese rinse

3 types of water—1) processing water
2) cleaning water
3) sanitary water

3 steps in CIP—1) pre rinse
2) caustic wash
3) final cold rinse

Waste processing—1) discharge treatment
2) waste disposal
3) onsite treatment

Then as is my first part of my project , analysis should be done to get a pictorial representation of the scheduling of water flow. The earlier waste water treatment methods are used to ensure that the proper levels of allowed discharge into the water bodies .As it is directly proportional to the amount of waste water being treated, the cost goes up with more water discharge. Hence it is considered wise to produce less waste water which needs a costly treatment. This can be possibly done by reducing the waste water production by using efficient methods scientifically or rather produce lesser quantities of the same and hence reduce fresh water consumption.
But ,the inherent discontinuous nature of batch processes complicates the development of a methodology to minimise wastewater, since the methodology not only has to take inlet and outlet concentrations into consideration, but also the times at which water is used during the operation unlike the continuous.
Its of major importance to analyse the process and classify them as continuous or batch and adopting the correct set of time point system like RTN or STN and make use of the appropriate insight based techniques or the mathematical modelling technique or a combination of both as is attempted in this work, where pinch analysis, a insight based technique is tried for its use as a heuristic for getting a good enough initial solution to optimise further. Minimisation of water can be done by initially scheduling or using a fixed schedule and then minimising or both can be done simultaneously. Due to complexity of the model a MINLP model arises, but in this work where modelling has been done, the non-linear terms have been neglected in the objective function.

The project at hand is of a cheese plant (Arla, Denmark) where water minimisation is desired. It has 3 cheese lines and an RO plant where the input whey or the initial waste water is sent for further purification and reuse for further usage.

Planning schedule:

Lead time – monad to moday
Plan – Tuesday to Monday
Transportation plan – Wednesday to Monday
Master plan Sunday to moday (more than a week)

It can be approached primarily by organising, analysing and interpreting the data collected and drawing inferences and conclusions like the optimum tank capacity in this project. After the gantt plots of the different schedule has been plotted , a rough idea of the relevance of the different models to choose the appropriate time representation (continuous or discrete) , material balances(STN or RTN), event representation like global time points(discrete or continuous) or unit specific or time slots or precedence relations. Then mathematical modelling optionally with heuristics is developed accordingly.

This research is part of the REWARD project, a large interdisciplinary research project, combining real-time monitoring, sensor development and quality and safety guidelines in pro-active industrial water management funded by the Danish Council for Strategic Research.

A brief project outline with a 3-fold objective:

1) Analysing the data (for a time span of 7 days) of a cheese process plant which involves:

a)producing Gantt charts of the schedule of each batch process of the different cheese lines

b)plots of the fresh water consumption assuming a discrete water flow

c)plot of whey(waste) input on a time horizon assuming a function for flow in post-processing(purifying)

d) calculating the amount of water consumption during each CIP run and plotting in Tableau

This part is important to get a sense of the raw date log files to graphically represent the batch schedule after manipulating the data and constraining the with small snippets of codes. Also it is used to find out the minimum buffer storage capacity of the tank.

2)Mathematical modelling for scheduling of a multipurpose plants that has both batch and continuous processes, based on the resource-task network (RTN) , global time point continuous representation, for different scenarios of cleaning and storage. It also involves post processing of plotting the gantt charts.

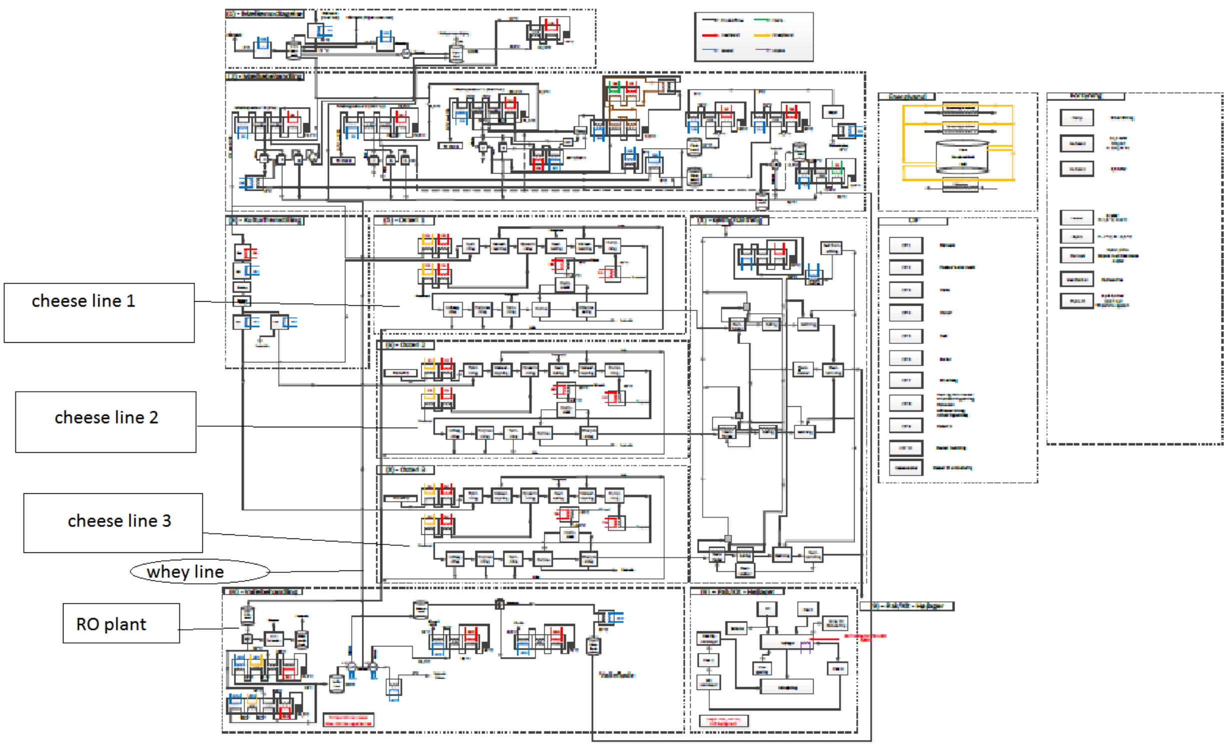
A skeleton model without a water flow and with only the various resources and tasks was created initially and a water layer of storage and cleaning has been added to this primary framework. It has incorporated both , the continuous and the batch models in a unified framework.

3)Analysing the possibility of applying pinch analysis as a heuristic for implementing in the scheduling model.

1)Analysing the data and finding the efficiency of water for storage (for a time span of 7 days) of a cheese process plant:

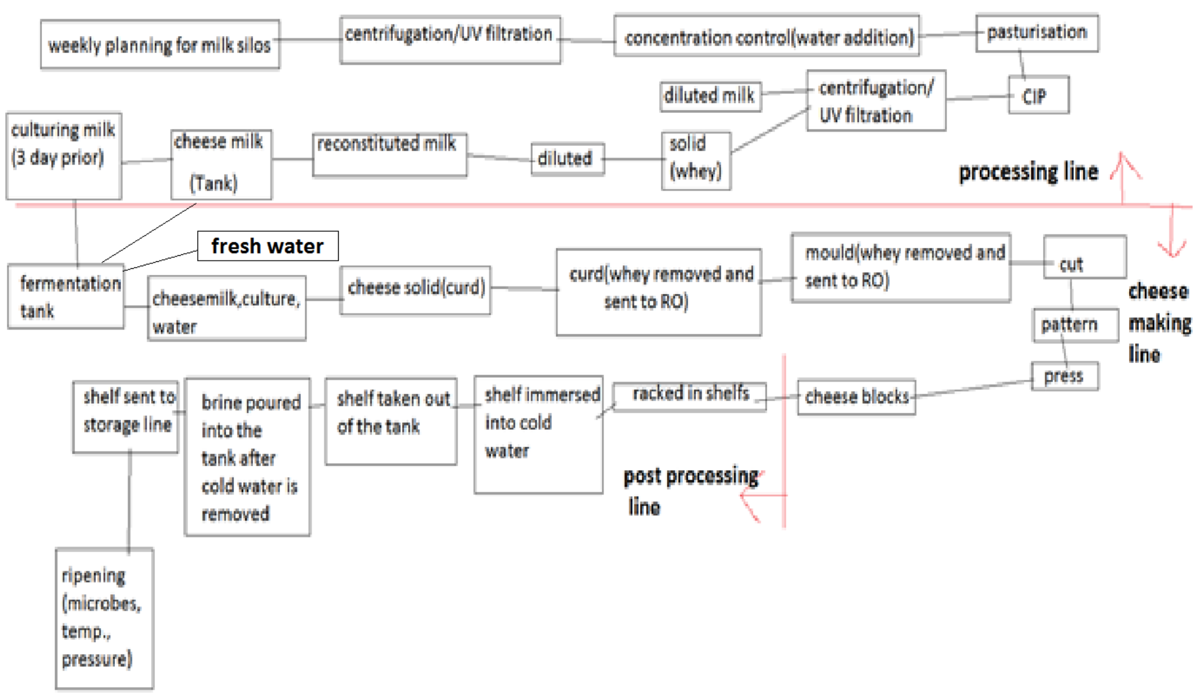
The following is the process plan of the cheese plant which is being analysed.

THE FACTORY LAYOUT OF THE ARLA PLANT:



The layout of the project includes three cheese lines, a RO plant and 10 Cleaning in Process units which are used to clean the units.

DIFFERENT PROCESSES IN THE ARLA CHEESE PLANT:



1a)Producing Gantt charts of the schedule of each batch process of the different cheese lines:

Producing gantt charts with the overall data available between different quantities on the y axis with respect to time gives a basic idea of how the processes are occurring on a uniform time scale and also to get a rough idea whether the solution generated in the mathematical model is in line with the actual schedule.

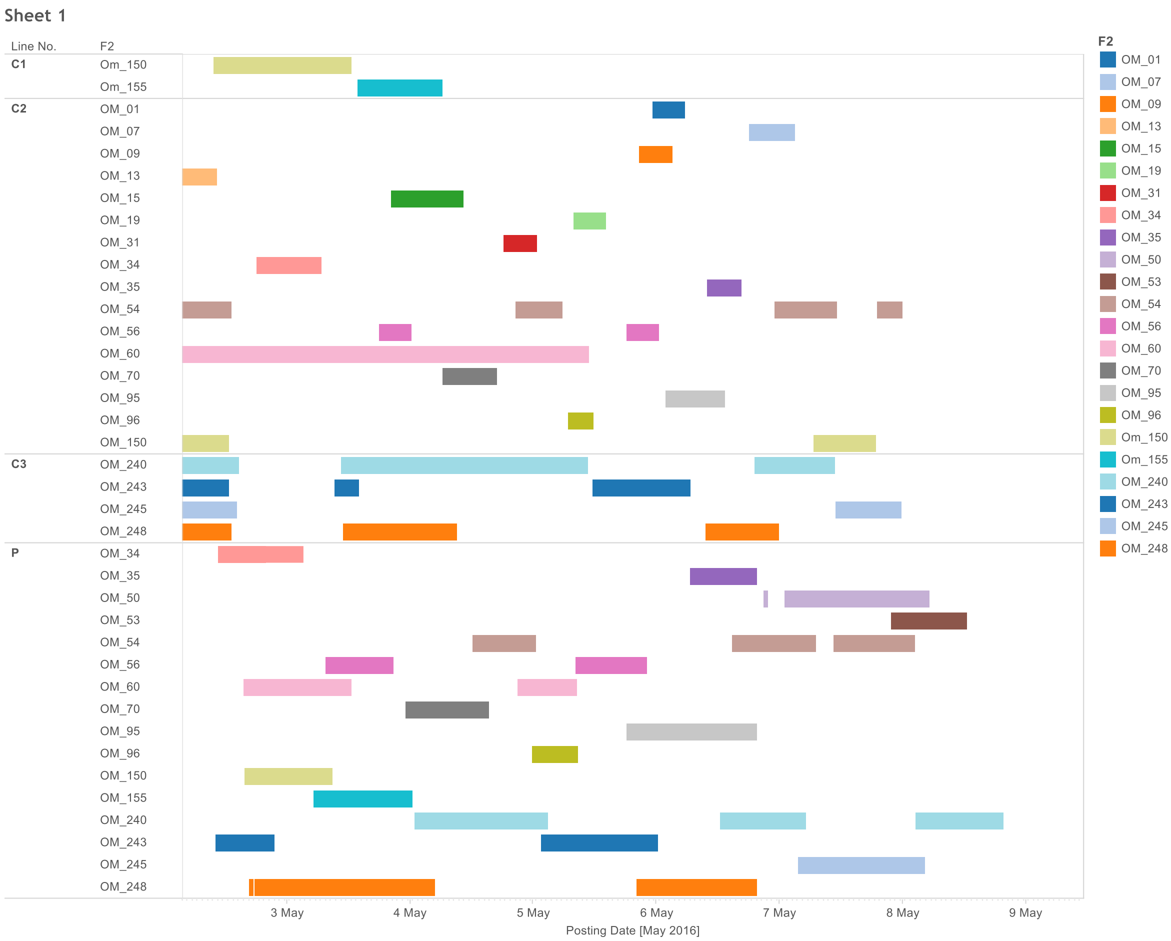
Also ,because data generally available in the real time would not be ideal, hence plotting such charts by combining all the relevant data from different files gives an idea about such a deviation as this cannot be interpreted from just the log of the start and end dates.

Different excel files which had all the necessary information were clubbed together by using the Vlookup function.

Calculation for plotting the gantt chart on a uniform horizon:

(DATEDIFF('second',[Log Open Date],[Log Close Dt]))/86,400

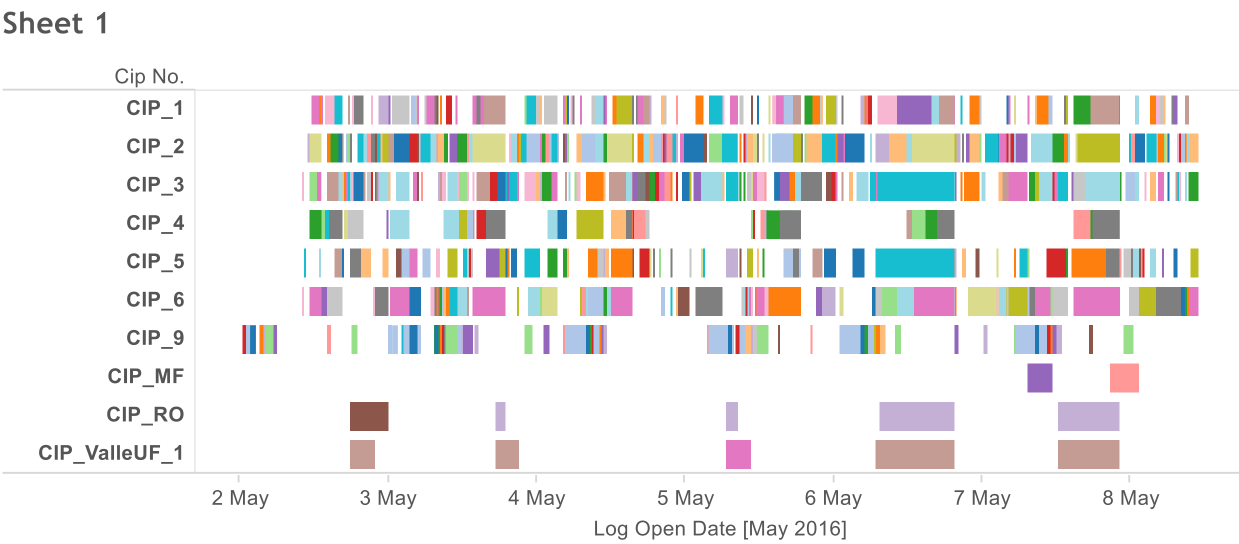
For each cheese line:



Posting Date for each F2 broken down by Line No.. Color shows details about F2. Size shows sum of Calculation1.

As can be seen above, Cheeseline 2 has lot of processes scheduled throughout the week, whereas as cheeseline 1 is not .This can be rectified by redistributing the processes in cheeseline 2 into chesseline 1 which has only 2 type of raw materials input to produce only 4 campaigns running 1 in OM150 and 3 in OM155.This is possible because line 1 is free after 4th May.If the cheesline 1 has a limitation of handling only select raw materials, the process should be made dynamic where different raw materials can be handled by changing a set of parameters.

For all the cip lines:

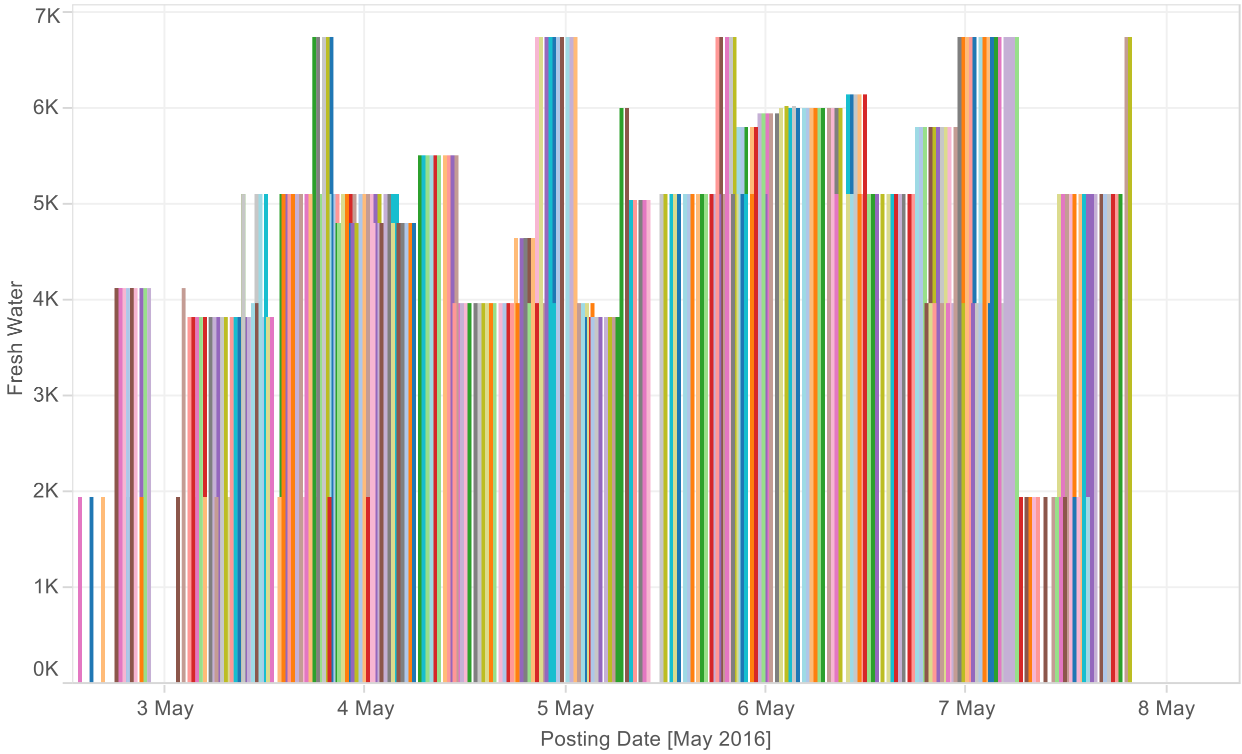


Log Open Date for each Cip No.. Color shows details about Recipe Id. Size shows sum of Calculation1.

1 b)Plots of the fresh water consumption assuming a discrete water flow:

After importing the excel file containing the fresh water consumption, for all the processes, it is plotted as a bar graph:

Sheet 1



The plot of sum of Fresh Water for Posting Date. Color shows details about Production Request Id.

As can be seen from the fresh water and whey graph, 2nd May,7th May has the least consumption and production of water and the remaining days more or less have the same usage of fresh water. It would be better if 2nd and 7th May has more processes from the processes planned for the other day, so that the storage or handling of fresh water capacity can be reduced and hence cost savings. Load on 3rd may could be reduced by evenly distributing the processes to the other days.

day in may	fresh water used
7	251,700
6	5,368,004
5	5,192,770
4	5,836,267
3	7,459,441
2	3,598,607

1 c)Plot of whey(waste) input on a time horizon assuming a function for flow in post-processing(purifying):

Function assumed:

F(t) - Quantity of whey that is given as input for the unit that is purified as RO water(whey is generated and given as input to RO plant at the rate of T=2hours per process)

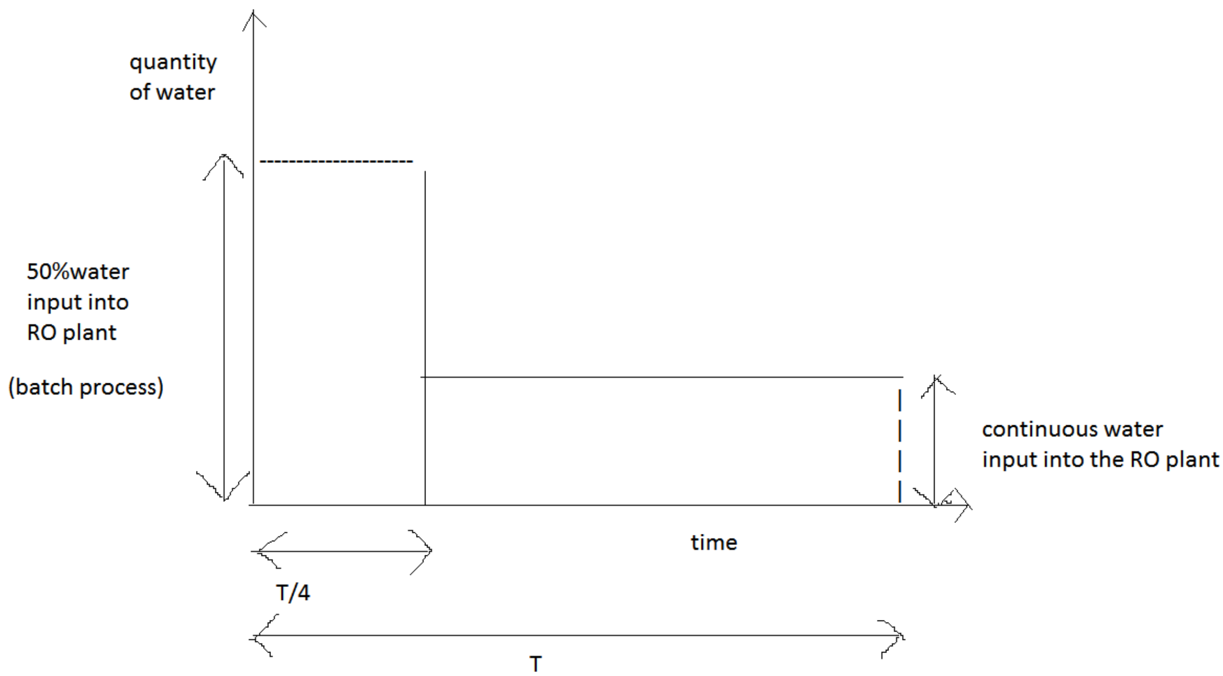
Q- total quantity of whey produced in the whole process

T is split into 1 time point/minute (discretised)

F(t)=0;t<T/4

=Q/2;t=T/4

=Q/180;T/4<t<=T

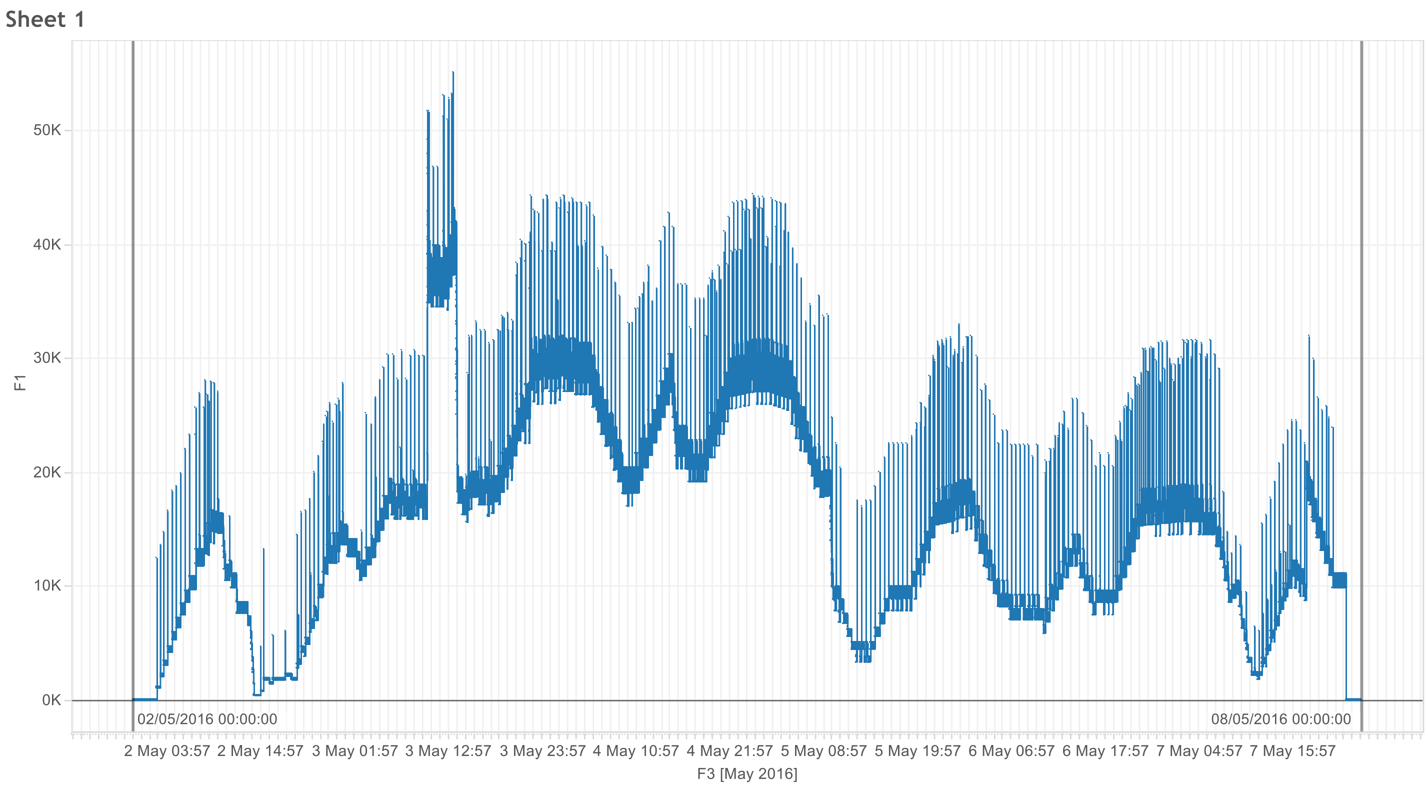


The reason for taking the above form of input is because initially fermentation takes place in any campaign of batch processes, hence till T/4,no whey is produced. At t=T/4,similar to a batch process ,whey is output which constitutes 50% of the total water or whey that comes out of the processes. After that from t=T/4 till T water is given as output to the RO plant as a continuous process.

After calculations in excel using “Vlookup” function, the whey quantity is calculated for each of the time points defining and breaking the function into different cases.(at T/4, T/4 to T).

For accommodating the different time horizons of each process into a unified scale, each log date is assigned a unique number equivalent and is imported into Matlab for further processing of the data:
(Appendix 1)

This csv file is then plotted on “Tableau” to generate the overall whey input vs time for the concerned length of 7 days for the different cheese lines.



The trend of sum of F1 for F3.

As can be seen from the above graph, whey production is the highest on 3rd may corresponding to the highest usage of fresh water.

As the quantity follows the following equation:

Raw milk-cream=cheese milk
Cheese milk+ fresh water +culture milk +miscellaneous-cheese=whey,

A direct correlation can be seen between fresh water and whey production more or less with fresh water being input at the start of the batch under the particular campaign but whey is input following the function assumed here.

Similarly for each of the 3 cheese lines having different set of batches the following code and the likes would be useful:
(Appendix 2)

The buffer storage capacity of tanks in the RO plant which gets the input as whey is having a maximum capacity already considered in the plant on a 20 hr. period. According to the above results from the data being analysed and from the matlab code, it has been found that for a one hour buffer, for storing 20hrs. of water 77.5 tonne capacity tank is required.
This was done by calculating the 20hr total consumption with different staring points starting from the local maximum for a considerable period with adjacent points also being more or less the same water flow rate.
(Appendix 3)

1 d) Calculating an approximate amount of water consumption during each CIP run and plotting in tableau:

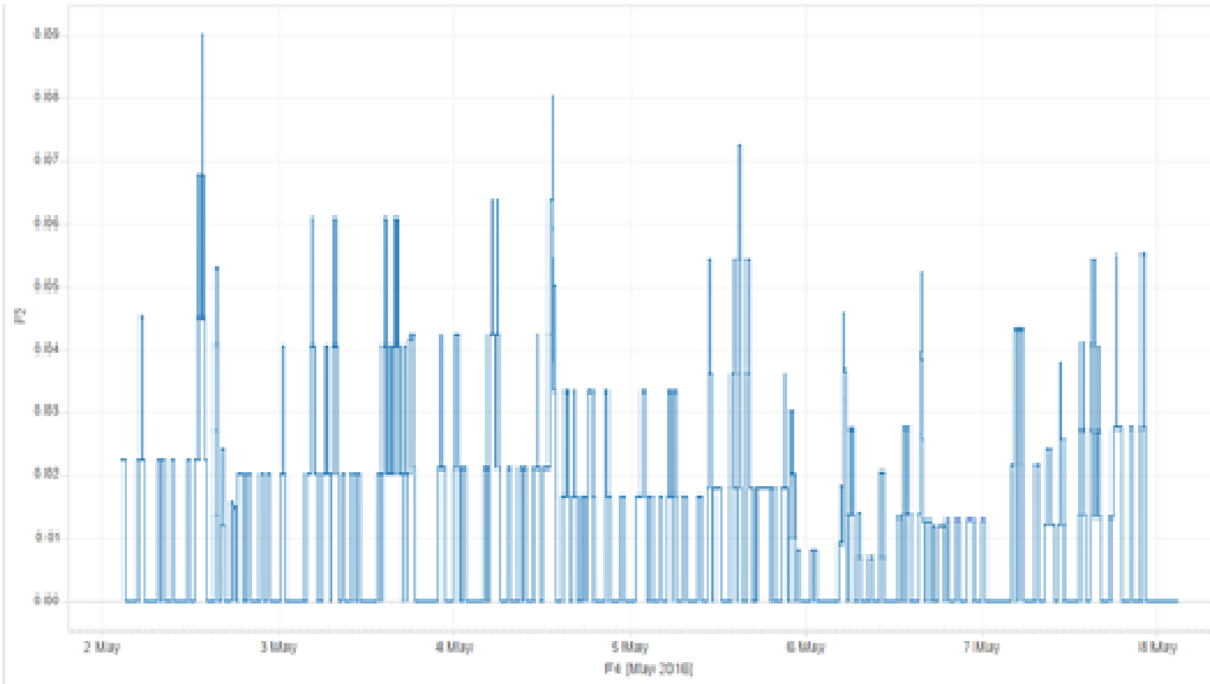
	Water consumption for the week
CIP_1	658.5734154
CIP_2	956.3107768
CIP_3	1,116.508535
CIP_4	849.6891948
CIP_5	649.7761057
CIP_6	149.0969387
CIP_8	12.02448243
CIP_9	333.6409185
CIP_10	1,106.304243

From the data it can be inferred that water consumption for CIP was highest for Cip_3 which is for valle (whey) and least water is used for Cip_9 which is for frame washing.

Mapping of the CIP logs was done. Certain calculations, using average consumption patterns were used to calculate an approximate amount of water consumption during each CIP run.

Assuming a constant function, the data is plotted in Tableau. To plot on a uniform time horizon, a scale was generated with each second having a unique general value equivalent of the date time format.
The discretisation of the time horizon is done by,

F2=actual consumption*502,524/duration of each run*58,163



2) Mathematical modelling (based on Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes,Pedro M. Castro et al.,2004) : Software Tools used:Cplex,Matlab,Excel and Tableau

The relevance of choosing this model for the project:

The following model was chosen because from the tableau plotted, it can be seen that a continuous and batch processes together can be represented in a single mathematical formulation like in this paper. This paper had a similar process but for a different application whose relevance has been exploited in the cheese line scheduling model. Here different types of raw materials are continuously mixed in mixer units with certain intermediates in between them. It is then packed in different product lines to produce

the final products. This constitutes the continuous process which is same as the campaigns run on the 3 cheese lines. The whey that is output for converting it into useful RO water in whey handling has both the batch process of a sudden water in flow at the 50%mark of the process and a constant continuous water in flow in the RO plant for the rest of the time in a continuous manner. Also both the scenarios involve cleaning-CIP (Cleaning In Process run)in the cheese line model and cleaning of the packing line in the model considered in this research paper. Also the global time point being used in the paper considered is useful in the cheese plant also because by using global time intervals, calculation for each point on the time horizon line need not be done which though might give a better solution would become tedious in a large scale real world problem like the cheese, line that is being considered here.
Also as both the processes involve a lot of resources and intermediates and raw materials and also keeping a track of these resources is of significance RTN implementation to this project is of great relevance.
Also the characteristic curve that was used for the whey production has a curve that has both the batch process at t=T/4 and continuous curve from T/4 to T.

Hence the formulation uses a RTN,uniform time grid representation and also a mixed integer linear programming model is used to solve for a optimal number of event points.

The basis and relevance of selecting the framework that has been chosen for this model from the literature:

Castro et al.,-An Improved RTN Continuous-Time Formulation for the Short-term Scheduling of Multipurpose Batch Plants.It has been shown that the STN-based continuous-time scheduling formulation of Ierapetritou and Floudas ,1998 is less accurate, as it violates time horizon constraints in certain cases.

Maravelias & Grossmann 2003 –In this work, the approach used in practically all continuous STN models, started with a small number of intervals and then the number of intervals was iteratively increased by one until there was no improvement in the objective function for a fixed number of iterations. Global time point approach for scheduling batch process plants disucssed in this paper was also utilised in this project.

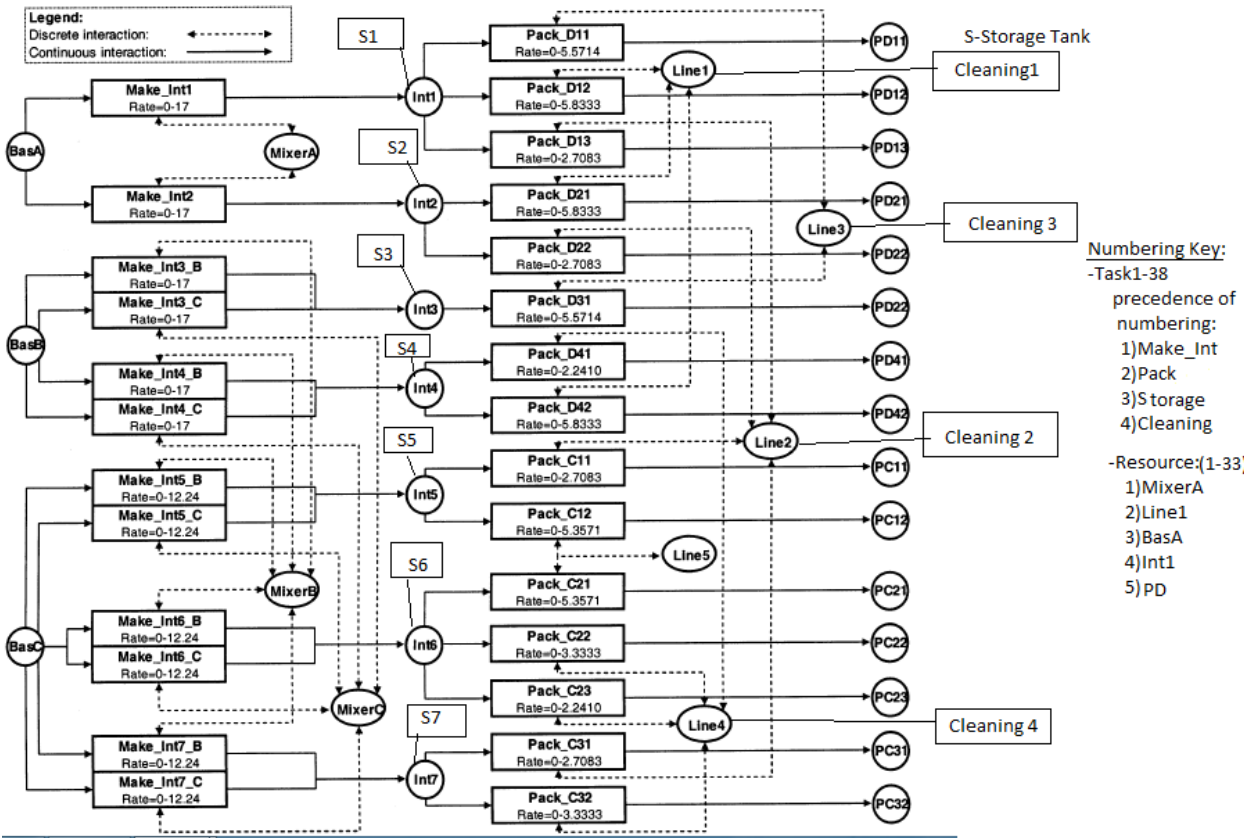
Iiro Harjunkoski et al, -It employed the RTN process representation since the unified treatment of resources allows for a focus on fewer sets of constraints. In a case study of the scheduling model of the Dow Chemicals, usage of RTN model led to low cost solutions ,cost reduction by waste reduction and the all the treatment units were used to their fullest efficiency.

Munawar A. Shaik et al., 2007, -In contrast to the STN representation, all the resources such as materials, processing and storage equipment, and utilities are uniformly treated in the RTN representation. STN model often leads to storage violations in the Gantt chart in real time, unless one considers storage as a separate task. The proposed RTN formulation overcomes this intricacy with the additional sequencing constraints. Compared to the STN-based model, the proposed RTN-based model gives better LP relaxation (due to the tightening constraint used), and even though the RTN version has more continuous variables and constraints, it solves relatively faster in some instances and it performs equally well in other instances of the examples considered for the case of unlimited storage capacity.

Mendez et al. 2006-the RTN-based formulations employ a uniform treatment and representation framework for all available resources through the idea that processing and storage tasks consume and release resources at their beginning and ending times. The major advantage of the RTN formulation over the STN counterpart arises in problems involving identical equipment. Here, the RTN formulation introduces a single binary variable instead of the multiple variables used by the STN model and hence complexity is reduced in the RTN model.

The different scenarios considered for implementation:

There are 4 different scenarios considered which include all combination of the presence and absence of cleaning task and storage task. These 4 different scenarios are considered for drawing conclusions on whether the presence of an optimal amount of storage increases the profit and whether it is more than compensated due to the investment in the storage facilities and also to calculate the increased cost of cleaning. Apart from this an optimum schedule of each continuous and batch (cleaning) processes is also determined.
4 cleaning and 7 storage tasks are implemented apart from the processes in the paper.



2.1)Nomenclature:

Sets/Indices:

C, *c*) equipment conditions
I, *i*) process tasks
I b) batch tasks
I c) continuous tasks
I emr) tasks that must exceed a certain minimum rate
*I*c) tasks belonging to condition *c*
I s) storage tasks
I ZW) tasks subject to zero-wait policies
R, *r*) process resources
*R*cl) equipment resources requiring cleaning
*R*eq) equipment resources (storage tanks not included)
*R*int) resources corresponding to intermediate products
*R*PK) equipment resources corresponding to packing lines) event points

Parameters:

Rro) initial amount of resource *r*

H) time horizon
cif) cost of executing task *i* (fixed cost term)
civ) cost of handling a unit amount of material in task *i*
(variable cost term)
cr) value of resource *r*
cjr) cost of holding a unit amount of inventory of resource

r over a unit period of time
Rr max) maximum availability of resource *r*
Rr min) minimum availability of resource *r*
Vi max) maximum amount of material processed by task *i*
Vi min) minimum amount of material processed by task *i*

delta*t*) maximum number of event points allowed between the beginning and end of a given task
delmin) minimum interval length
lambda) coefficient for the rate of generation of resource *r* by task *i*
mu) coefficient for the binary extent of resource *r* at the start of task *i*
mubar) coefficient for the binary extent of resource *r* at the end of task *i*
Rhomax) maximum allowable rate of task *i*
Rhomin) minimum allowable rate of task *i*
Variables
*N*bar) binary variable that assigns the end of task *i*, which began at *t*, to point *t*¢
R) excess amount of resource *r* at event point *t*
T) absolute time of event point *t*
zetabar) total amount of material processed by task *i*, starting at *t* and finishing at *t*¢

2.2)Tuple used to store 3-D data in Excel:

The cplex output of Nbar is stored in the excel as a tuple.
initialise a tuple for the 3D variable Nbar as sheetwrite can’t be used directly.
(Appendix 4)

2.3)MATLAB code for interfacing Cplex and Tableau:

Following is the the case of No Storage:
After importing the cplex decision variable Nbar into Matlab, the following code is executed to eliminate Nbar values of 0 and choose Nbar value of 1 and corresponding time point numbers are changed to the actual time.

(Appendix 5)

The above file written in the csv format is used in Tableau to plot the gantt chart.

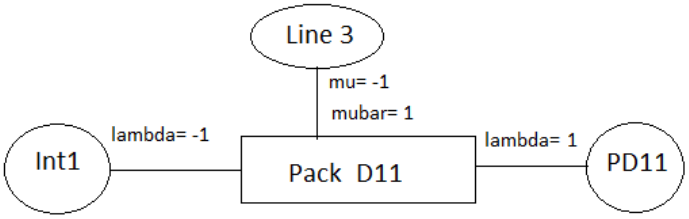
2.4)Cplex code and tableau plot of the output in cplex:

Time horizon(H)=120 hrs.

No. of time points considered = 8

Assumption: no cost for cleaning and storage considered in the profit.

lambda,mu,mubar values should be defined as given below for any set of task, resource:

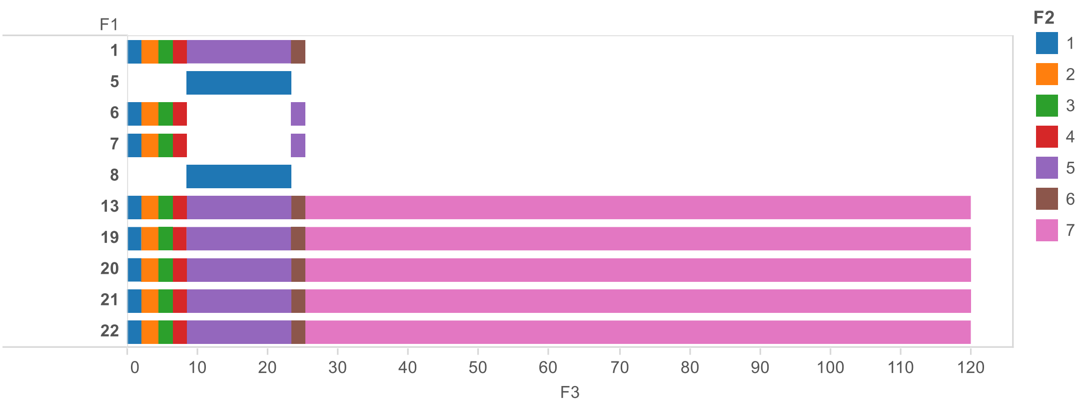


2.4.1)Without storage and cleaning:

(Appendix 6)

Tableau plot:

Sheet 1

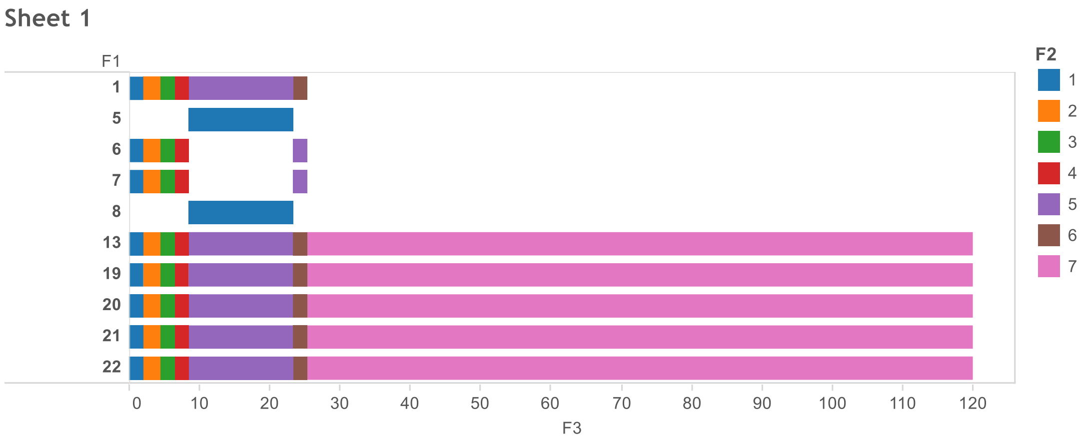


Sum of F3 for each F1. Color shows details about F2. Size shows sum of Calculation1.

2.4.2)With storage:

(Appendix 7)

Tableau plot:

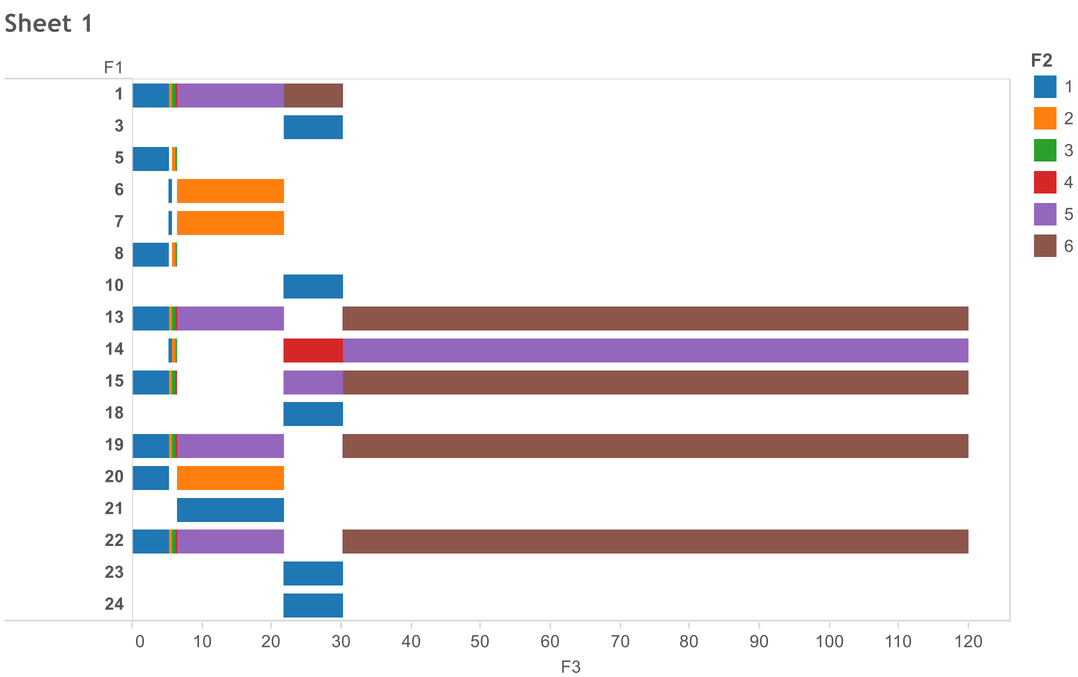


Sum of F3 for each F1. Color shows details about F2. Size shows sum of Calculation1.

2.4.3) Without storage:

(Appendix 8)

Tableau plot:



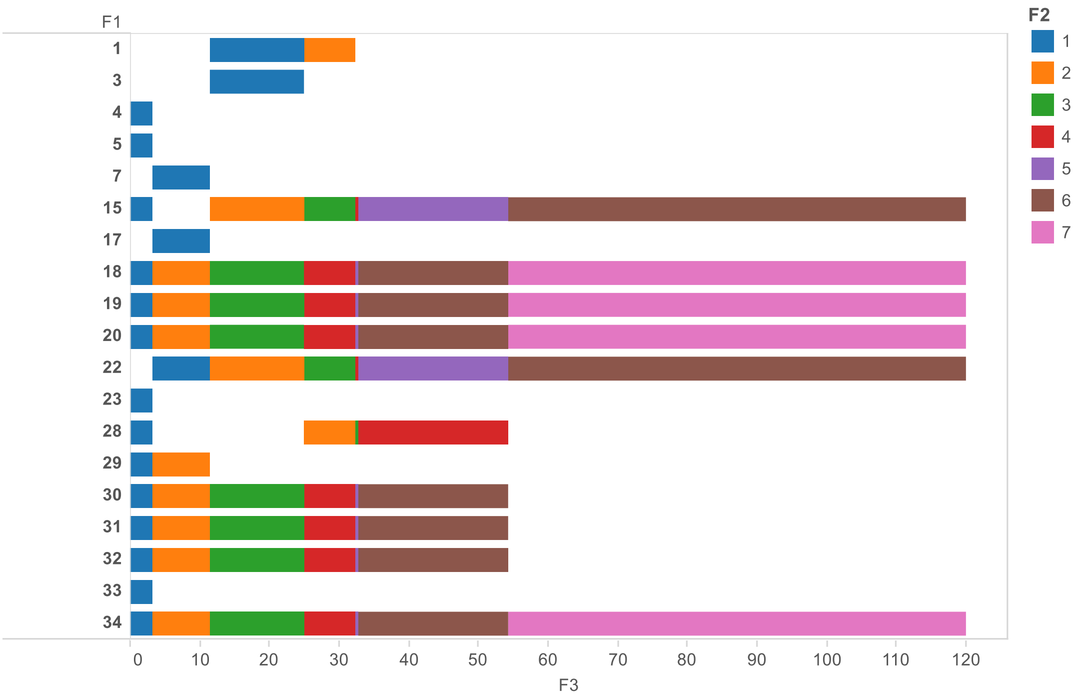
Sum of F3 for each F1. Color shows details about F2. Size shows sum of f.

2.4.4)With cleaning and storage:

(Appendix 9)

Tableau plot:

Sheet 1



Sum of F3 for each F1. Color shows details about F2. Size shows sum of Calculation1.

Overall comparison:

	Profit	Event points	Constraints	Continuous variables	Binary variables
without storage and with cleaning	22,381.77	8	1616	462	217
without storage and without cleaning	22,583	8	1536	462	189
with storage and without cleaning	23,274	8	1578	462	238
with storage and with cleaning	64,587.5	8	1774	490	266

3)Analysing the possibility of applying pinch analysis as a heuristic for implementing in the scheduling model.(Literature - "A new graphical targeting method for water minimisation,N. Hallale,2001" and conclusion):

3.1)Traditional approach based on pinch analysis for heat integration:

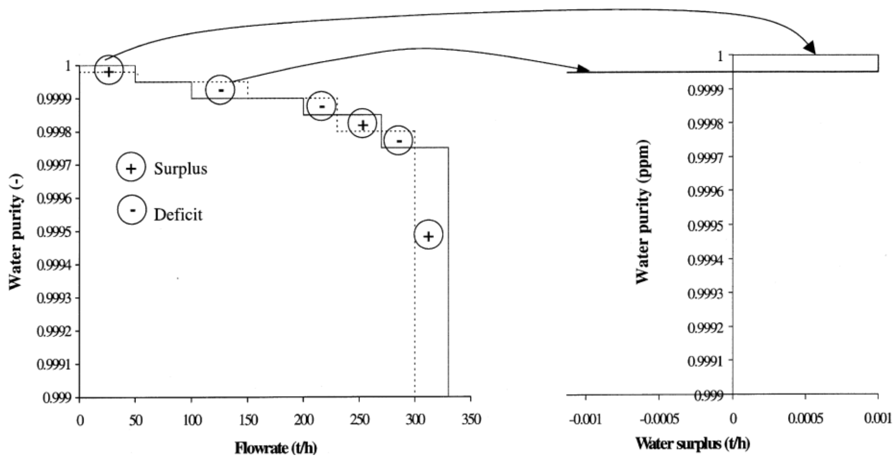
3.1.1) The method:

Fresh water and wastewater can be minimised by maximising the reuse of water within a process plant. Wang and Smith 1994 presented a graphical methodology to target the minimum water and wastewater flowrates. This approach was based on an extension of the pinch analysis techniques for heat integration. It considers each water-using operation as being described by the mass transfer of a certain contaminants. from the process itself to the water stream. By specifying the maximum allowable inlet and outlet contaminant concentrations for each operation, a limiting profile can be constructed . These are combined to form a limiting composite curve.against which a water supply line can be matched. The inverse of the slope of this line gives the target for the minimum fresh water and wastewater. The point that limits the slope of the line has been termed the water pinch. Note that this method is based upon a plot of contaminant concentration vs. mass load. Wang and Smith 1994. also presented a design procedure, which allowed the targets to be met. Dhole et al. 1996. noted that the pinch divided the problem into two regions: that above the pinch and that below it. In order to achieve the targets, fresh water should not be used below the pinch and also, sources above the pinch should not be discharged as
wastewater.

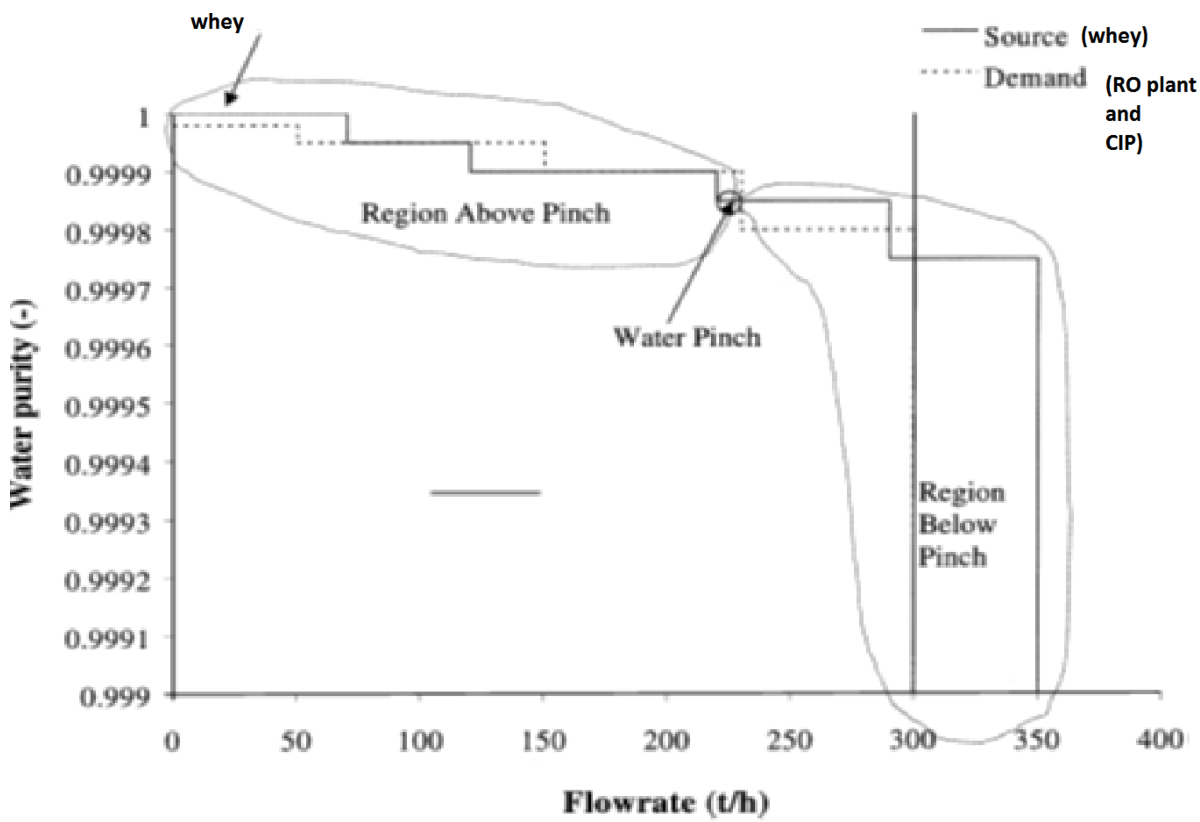
3.1.2) Drawbacks:

However, this representation unfortunately does not give a clear picture of the situation. This is because mixing of water sources can change the shape of the source composite and hence the targets. Mixing of two sources results in a new source at some blended concentration. This allows greater overlap between the composites and also changes the pinch point.
Thus, the targets given by this approach cannot be considered as true targets as they depend on the mixing pattern, which is actually part of the network design. In fact, careful analysis of the article by Dhole et al. 1996. shows that the composite curves are only constructed after a network design has been developed using mathematical programming. In other words, the curves give a graphical representation of a particular design. A true targeting approach would predict a minimum water flowrate ahead of any design.

3.2) Water Surplus Diagram:



The drawbacks of the previous method lead to this new method, the Water Surplus Diagram.

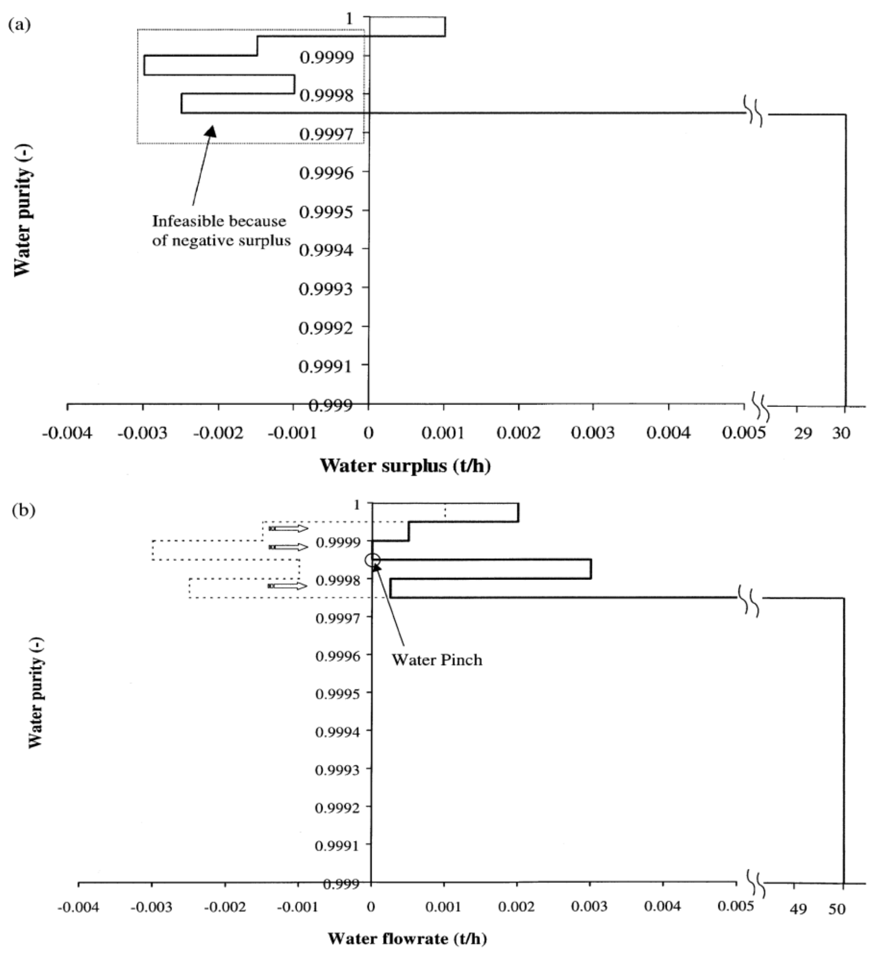


3.3) Procedure to find the minimum fresh water(water surplus diagram):

Notice that with the assumed value of fresh water flowrate, part of the plot lies in the negative region. This means that there will not be sufficient water purity at all points in the network and so the second feasibility criterion is not satisfied. Thus, more fresh water must be added, until no part of the surplus diagram is negative. The *minimum* fresh water target will be the flowrate that causes the surplus diagram to just touch the vertical axis. This is determined by repeating the procedure described above for different flowrates. This can be carried out by hand, but using a spreadsheet or a simple computer code will avoid the tediousness of the repetitive calculations.
These guidelines may be used to design a network manually if desired.

3.4) Mathematical programming as an alternative:

However, it is convenient to use a mathematical program for design. With fixed source purities, the problem is a linear program LP. Which can be solved using well-established methods that guarantee a global optimum Vaidyanathan et al., 1998.



From the previous section, it should be apparent that the minimum fresh water and wastewater flowrates can actually be determined directly by using mathematical programming and that there is no need to go through the targeting procedure beforehand. This is true, especially since the mathematical program is linear and can, therefore, guarantee a global optimum.

3.5) Water surplus diagram vs mathematical programming:

However, the graphical tools developed in this paper have advantages in terms of offering increased insights into the problem. One area where this is very valuable is in the assessing of process modifications. Process modifications are changes made to the individual water-using operations to modify either the flowrates or concentrations of water sources or demands. A mathematical program can certainly be used to evaluate the effect of process modifications, but this requires a trial-and-error approach where various modifications are made and then analysed by running the mathematical program again. The pinch approach has an advantage in that the insights from the graphical representation give the designer some clear guidelines about which process modifications will be beneficial.

3.6) Conclusion and relevance of water surplus diagram to the cheese production model:

Pinch analysis is an Insight-Based Techniques for Water Minimization in Batch Processes.
Hence, it would be best to find the pinch point using the water surplus diagram as a initial solution (Heuristic) and follow the guidelines:

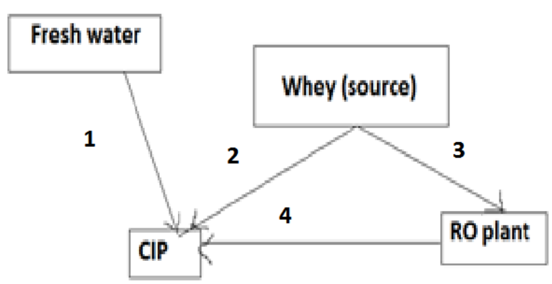
1)In order to achieve the flowrate targets determined above, it is necessary to observe the pinch division. This means that water sources above the pinch including fresh water may not feed demands below the pinch and may also not mix with sources that are below the pinch concentration.

2)When it is above the demand composite, this is a region with a surplus of pure water i.e. water is available with a purity greater than what is required. When it is below, this is a region with a deficit of pure water i.e. water is required with a purity higher than what is available.

both of which can be incorporated as constraints.

The pinch analysis becomes relevant and can be compared to the cheese line model.The source here being the whey and demand is the CIP line which needs water for cleaning and also the RO plant which needs water for purifying. Fresh water being available always.

In this case of the cheese line, source is formed by the whey output and for any iteration needed to alter the source curve such that there is no water deficit, fresh water is used to compensate initially such that the water surplus diagram curve always lies in the first quadrant only.
Hence after the fresh water needed for the CIP line is added, the curve would have water surplus except for the pinch point. This water could then be sent to the RO plant for purification. This way, the Cip can be done with a minimum amount of fresh water. Also if RO processing is cheaper than fresh water, the initial flow rate assumed can be lesser i.e., the initially assumed quantity of fresh water can be reduced and compensated by the RO processed water that takes input from the whey line and maintains the water surplus curve in the positive first quadrant. This depends on the trade off between the fresh water cost at optimum profitable level, the amount of water that needs to be given as input to the RO plant for usage in the CIP process .This makes sure that this whole process of reusing the regenerated whey water from the RO plant is profitable.
But in the case that whey is useful and valuable than the RO processing cost, line (2) will not be present and all the whey water goes to the RO plant and only then it goes to the CIP cleaning.In that case only fresh water is added to offset the negative portions of the water surplus curve.



Sorin and B´edard 1999. correctly observed that the approach of Dhole et al. 1996. (the conventional pinch analysis method based on heat integration method) actually defines many ‘local’ pinch points, but not necessarily the true or ‘global’ pinch point.To account for the different set of such processes in the actual cheese point which consists of a large number of such sets, attaining a global pinch point again becomes a difficulty.This is however overcome by the water surplus diagram.
The region above pinch point has no waste water generation and the region below the pinch point does not need fresh water.

4)CONCLUSION:

Fresh water consumption is unevenly spread throughout one week time horizon of concern. according to the current schedule of the Arla plant as is evident from the gantt charts plotted.

If this is evened out, the maximum capacity of the tank required would reduce which basically lies in altering the buffer time of the water tank. The current capacity as for the current plant interpreted from the data can even be reduced.

The chosen RTN, network and global time point system have proved to be effective in multiple identical equipment modelling and this model did not have any storage violation and the computation time and complexity are less. Even though the RTN version has more continuous variables and constraints, it solves relatively faster in some instances and it performs equally well in other instances of the examples considered for the case of unlimited storage capacity. Hence for all these reasons and the fact that a complex industry can be better represented with resources (even identical parallel ones) and equipments and processes separately justifies the suitability of this model for future research work on the same.

Also the whey production is maximum on one of the days disproportionately, hence scheduling the model effectively can even out the whey which can be sent to the regeneration plant and swiftly reused according to the corresponding contaminant concentration allowed.

Also there seems to be a lot of deviation from the planned timeline of execution in the Arla production schedule. This could cause problems as a plant even before the optimum possible schedule is not following the schedule and even if an optimum solution is developed, improper execution could cause cascading of water and overlapping of schedules. This could possibly be avoided by using the manufacturing system simulation which accounts for the real world uncertainties in the cheese plant.

Coming to the numerical interpretation and significance of the profits that are viable in the scenario run in this project suggests that incorporating storage containers increase the profit about three fold in comparison to the case where storage is not used. Though in both the above scenarios, cleaning is considered.

Also when storage is used the machines seem to be continuously running throughout the time horizon without having a lot of batches. This could possibly be because the number of time points considered here is 8, and more points could imply more batches instead of a predominantly continuous process throughout the timeline as it leads to the underutilisation of the parallel processes, in this case predominantly when two or three packing units are available for an intermediate (resource) preceding it only one packing unit (task) is utilised.

Appendix:

1)
```
for i=1:389
a(1,A(i,1))=A(i,2);
end
for j=1:389
for k=A(j,1):A(j,5)
a(1,k)=a(1,k)+A(j,4);
end
end
b=transpose(a);
csvwrite('mat4.csv',b);
```

2)
```
for i=1:36
a(1,A(i,1))=A(i,2);
end
for j=1:36
for k=A(j,1):A(j,5)
a(1,k)=a(1,k)+A(j,4);
end
end
b=transpose(a);
csvwrite('solution ch1mat.csv',b);
```

3)
```
>> q=zeros(3,269400);
for m=[125400:3600:269400]
for i=m:269400
j=ceil((i-125400)/72000)+1;
q(j,m)=q(j,m)+A(i,1);
end
end
```

```
mx=max(q);
x=max(mx);
for m=[125400:3600:269400]
for i=m:269400
j=ceil((i-125400)/72000)+1;
if q(j,m)==x
jmax=j;
mmax=m;
end
end
end
```

4)

```
 tuple NbarTuple {
int I1;
int T1;
int T2;
int value;}
{NbarTuple}NbarSet={<i,j,k,Nbar[i,j,k]>| i in I,j in T,k in T};

//size and range of the tuple.
string outputRange; // = "ScheduledTrips!A:L";
int numRows; // = Shipments.size;

execute {
  numRows = NbarSet.size;
  outputRange = "cplex1.2!A2:" + "A" + (numRows+1);
  writeln(outputRange);
}
```

5)

```
j=1;
for i=1:1728//1728 is the product of (no. of tasks ,(no. of time points)^2)

          if A(i,2)==1//1 is the time point
      a(i,2)=0;
   elseif A(i,2)==2
      a(i,2)=5.18;//5.18 is the time at which 2nd time point is present
          elseif A(i,2)==3
      a(i,2)=5.63;
          elseif A(i,2)==4
      a(i,2)=6.08;
   elseif A(i,2)==5
      a(i,2)=6.52;
          elseif A(i,2)==6
      a(i,2)=21.81;
         elseif A(i,2)==7
      a(i,2)=30.25;
   elseif A(i,2)==8
      a(i,2)=120;
          end
      if A(i,3)==1
      a(i,3)=0;
   elseif A(i,3)==2
```

```
      a(i,3)=5.18;
         elseif A(i,3)==3
      a(i,3)=5.63;
         elseif A(i,3)==4
      a(i,3)=6.08;
         elseif A(i,3)==5
      a(i,3)=6.52;
   elseif A(i,3)==6
      a(i,3)=21.81;
   elseif A(i,3)==7
      a(i,3)=30.25;
   elseif A(i,3)==8
      a(i,3)=120;
      end
 a(i,1)=A(i,1);


 if A(i,4)==1
    b(j,1)=a(i,1);
    b(j,2)=a(i,2);
    b(j,3)=a(i,3);
    j=j+1;
 end
end

csvwrite('cleaning1.4.csv',b);    //cleaning 1.4.csv is the file into which the output is written.
```

for the other cases(storage,storage with and without cleaning), it can be modified similarly.

6).**mod file**

```
/*********************************************
 * OPL 12.6.1.0 Model
 * Author: Karthik Sundaram
 * Creation Date: 18 Jul 2016 at 16:04:45
 *********************************************/
//Sets
int zC=...;
int zI=...;


int zIco=...;
int zIemr=...;

int zR=...;
int zIb=...;
int zReq=...;
int zRint=...;
int zRpk=...;
int zRcl=...;
int zT=...;
```

```
range C = 1..zC;
range Rcl=4..zRcl;
 range Ib=28..zIb;
 range I = 1..zI;


    range Ico = 1..zIco;
     range Iemr = 1..zIemr;
      range R = 1..zR;

        range Req = 1..zReq;
         range Rint = 12..zRint;
         range Rpk = 4..zRpk;
         range T = 1..zT;




//Parameters
int H=...;
float Cif[Ico]=...;
float Civ[Ico]=...;
float Cr[R]=...;
float Rmax[R]=...;
float Rmin[R]=...;
float Vmax[I]=...;
float Vmin[I]=...;

int delt=...;
float deltmin=...;

float xyz[C,I]=...;
float pqr[Rcl,I]=...;
float lambda[R,I]=...;
float mu[R,I]=...;   //start
float mubar[R,I]=...; //end
float nu[R,Ico]=...;  //start
float nubar[R,Ico]=...;  //end
float rhomax[Ico]=...;
float rhomin[Ico]=...;
float Rro[R]=...;
int lastT=...;

//Decision variables
dvar boolean Nbar[I,T,T];
dvar float+ Rrt[R,T];
dvar float+ Tt[T];
dvar float+ zetabar[I,T,T];

constraint ct1[T,T,Ico];
constraint ct2[T,T,Ico];
constraint ct5[I,T];
constraint ct6[I,T];
constraint ct7[R,T];
constraint ct8[R,T];
```

```
constraint ct10[R,T];

constraint ct12[I,T];

constraint ct14[I,T];
constraint ct15[R,T];
constraint ct16[R,T];

dexpr float profit =(sum(r in R)(Cr[r]*(Rrt[r,lastT]))-
sum(i in Ico,t in T:t!=lastT,tp in T:t<tp&&tp<=t+delt)
  (Cif[i]*Nbar[i,t,tp]+Civ[i]*zetabar[i,t,tp]));


maximize (profit);

subject to
{
//Timing of tasks
//batch tasks

//zero wait tasks

//continuous tasks
forall(t,tp in T:tp==t+1&&t!=lastT,i in Ico)
{
ct1[t,tp,i]:Tt[tp]-Tt[t]>=zetabar[i,t,tp]/rhomax[i];
}
//continuous tasks that must exceed minimum processing rate
forall(t,tp in T:tp==t+1&&t!=lastT,i in Iemr)
{
ct2[t,tp,i]:Tt[tp]-Tt[t]<=H*(1-Nbar[i,t,tp])+zetabar[i,t,tp]/rhomin[i];
}

//Initialization
ct3:Tt[1]==0;
ct4:Tt[lastT]==H;

forall(i in Ico,t in T:t<lastT)
{
ct5[i,t]:deltmin*rhomin[i]*Nbar[i,t,t+1]<=zetabar[i,t,t+1];
ct6[i,t]:H*rhomax[i]*Nbar[i,t,t+1]>=zetabar[i,t,t+1];
}


forall(r in R, t in T)
{
if(t>1)
{
ct7[r,t]:Rrt[r,t]==Rrt[r,t-1]+sum(i in Ico:t<lastT)(mu[r,i]*Nbar[i,t,t+1]+
mubar[r,i]*Nbar[i,t-1,t]+lambda[r,i]*zetabar[i,t-1,t]);
}
else if (t==1)
{
ct8[r,t]:Rrt[r,t]==Rro[r]+sum(i in Ico:t<lastT)(mu[r,i]*Nbar[i,t,t+1]);
}
```

```
}
forall (r in Rint, t in T)
  {
 Rrt[r,t]==0;
  }
    forall(r in Rpk,t in T:t!=lastT)
   {
    ct10[r,t]:Rrt[r,t]==0;
}


//cleaning constraints
forall(r in Rcl,t in T:t<lastT-1,c in C)
  {
 (sum(i in I)xyz[c,i]*mubar[r,i]*Nbar[i,t,t+1])+(sum(i in I:i not in {1,2,3,4,5,6,7,8,9,10,11,12,22,23})
 (1-xyz[c,i])*mubar[r,i]*Nbar[i,t+1,t+2])<=1;
  }

forall(i in Ib,c in C,t in T:t>1 && t<lastT-2)
  {
sum(ibar in I)sum(r in Rcl)pqr[r,i]*xyz[c,ibar]*mubar[r,ibar]*Nbar[ibar,t-1,t]+
(sum(ibar in I:ibar not in {1,2,3,4,5,6,7,8,9,10,11,12,22,23})sum(r in Rcl)pqr[r,i]*(1-
xyz[c,ibar])*mubar[r,ibar]
*Nbar[ibar,t+1,t+2])<=1+Nbar[i,t,t+1];
  }

  forall(r in R,t in T)
{
ct15[r,t]:Rmin[r]<=Rrt[r,t];
ct16[r,t]:Rmax[r]>=Rrt[r,t];
}
}

tuple NbarTuple {
int I1;
int T1;
int T2;
int value;}
{NbarTuple}NbarSet={<i,j,k,Nbar[i,j,k]>| i in I,j in T,k in T};


string outputRange; // = "ScheduledTrips!A:L";
int numRows; // = Shipments.size;

execute {
   numRows = NbarSet.size;
   outputRange = "cplex1.2!A2:" + "A" + (numRows+1);
   writeln(outputRange);
}
```

.dat file

```
/*********************************************
 * OPL 12.6.1.0 Data
 * Author: Karthik Sundaram
```

```
 * Creation Date: 18 Jul 2016 at 16:05:06
 *********************************************/
SheetConnection File("cplex1.2newnew.xlsx");

 zC=2;
 zIb=31;
 zI=31;
 zIco=27;
 zRcl=7;
 zIemr=12;
 zR=33;
 zReq=8;
 zRint=18;
 zRpk=8;
 zT=8;



 H from SheetRead (File, "Sheet1!B1");
 Cif from SheetRead (File, "Sheet1!B4:B30");
 Civ from SheetRead (File, "Sheet1!E4:E30");
 Cr from SheetRead (File, "Sheet1!H4:H36");
 Rmax from SheetRead (File, "Sheet1!K4:K36");
 Rmin from SheetRead (File, "Sheet1!N4:N36");
 Vmax from SheetRead (File, "Sheet1!B142:B172");
 Vmin from SheetRead (File, "Sheet1!E142:E172");

delt from SheetRead (File, "Sheet1!E1");
deltmin from SheetRead (File, "Sheet1!H1");
lambda from SheetRead (File, "Sheet1!H39:AL71");
mu from SheetRead (File, "Sheet1!H73:AL105");
mubar from SheetRead (File, "Sheet1!H107:AL139");
nu from SheetRead (File, "Sheet1!H141:AH173");
nubar from SheetRead (File, "Sheet1!H175:AH207");
rhomax from SheetRead (File, "Sheet1!H209:H235");
rhomin from SheetRead (File, "Sheet1!K209:K235");
lastT from SheetRead (File, "Sheet1!K1");
Rro from SheetRead (File, "Sheet1!B71:B103");

xyz from SheetRead (File, "Sheet1!B337:AF338");
pqr from SheetRead (File, "Sheet1!B341:AF344");
NbarSet to SheetWrite(File,"Sheet3!A1:D1984");
```

7) **.mod file**

```
/*********************************************
 * OPL 12.6.1.0 Model
 * Author: Karthik Sundaram
 * Creation Date: 22 Jun 2016 at 10:51:50
 *********************************************/
//Sets
int zC=...;
int zI=...;
```

```
int zIs=...;

int zIco=...;
int zIemr=...;
int zRIs=...;
int zR=...;

int zReq=...;
int zRint=...;
int zRpk=...;

int zT=...;

 range C = 1..zC;
  range I = 1..zI;
   range RIs=12..zRIs;
     range Is=28..zIs;
     range Ico = 1..zIco;
      range Iemr = 1..zIemr;
       range R = 1..zR;

         range Req = 1..zReq;
          range Rint = 12..zRint;
          range Rpk = 4..zRpk;
          range T = 1..zT;




//Parameters
int H=...;
float Cif[Ico]=...;
float Civ[Ico]=...;
float Cr[R]=...;
float Rmax[R]=...;
float Rmin[R]=...;
float Vmax[I]=...;
float Vmin[I]=...;
float a[Ico]=...;
float b[Ico]=...;
int delt=...;
float deltmin=...;
float abc[RIs,Is]=...;
float lambda[R,I]=...;
float mu[R,I]=...;   //start
float mubar[R,I]=...; //end
float nu[R,Ico]=...;  //start
float nubar[R,Ico]=...;  //end
float rhomax[Ico]=...;
float rhomin[Ico]=...;
float Rro[R]=...;
int lastT=...;

//Decision variables
dvar boolean Nbar[I,T,T];
dvar float+ Rrt[R,T];
```

```
dvar float+ Tt[T];
dvar float+ zetabar[I,T,T];

constraint ct1[T,T,Ico];
constraint ct2[T,T,Ico];
constraint ct5[I,T];
constraint ct6[I,T];
constraint ct7[R,T];
constraint ct8[R,T];

constraint ct10[R,T];

constraint ct12[I,T];

constraint ct14[I,T];
constraint ct15[R,T];
constraint ct16[R,T];

dexpr float profit =(sum(r in R)(Cr[r]*(Rrt[r,lastT]))-
sum(i in Ico,t in T:t!=lastT,tp in T:t<tp&&tp<=t+delt)
  (Cif[i]*Nbar[i,t,tp]+Civ[i]*zetabar[i,t,tp]));


maximize (profit);

subject to
{
//Timing of tasks
//batch tasks

//zero wait tasks

//continuous tasks
forall(t,tp in T:tp==t+1&&t!=lastT,i in Ico)
{
ct1[t,tp,i]:Tt[tp]-Tt[t]>=zetabar[i,t,tp]/rhomax[i];
}
//continuous tasks that must exceed minimum processing rate
forall(t,tp in T:tp==t+1&&t!=lastT,i in Iemr)
{
ct2[t,tp,i]:Tt[tp]-Tt[t]<=H*(1-Nbar[i,t,tp])+zetabar[i,t,tp]/rhomin[i];
}

//Initialization
ct3:Tt[1]==0;
ct4:Tt[lastT]==H;

forall(i in Ico,t in T:t<lastT)
{
ct5[i,t]:deltmin*rhomin[i]*Nbar[i,t,t+1]<=zetabar[i,t,t+1];
ct6[i,t]:H*rhomax[i]*Nbar[i,t,t+1]>=zetabar[i,t,t+1];
}


forall(r in R, t in T)
{
```

```
if(t>1)
{
ct7[r,t]:Rrt[r,t]==Rrt[r,t-1]+sum(i in Ico:t<lastT)(mu[r,i]*Nbar[i,t,t+1]+
mubar[r,i]*Nbar[i,t-1,t]+lambda[r,i]*zetabar[i,t-1,t])+
sum(i in Is:t<lastT)(mu[r,i]*Nbar[i,t,t+1]+mubar[r,i]*Nbar[i,t-1,t]);
}
else if (t==1)
{
ct8[r,t]:Rrt[r,t]==Rro[r]+sum(i in Ico:t<lastT)(mu[r,i]*Nbar[i,t,t+1]);
}
}

   forall(r in Rpk,t in T:t!=lastT)
  {
   ct10[r,t]:Rrt[r,t]==0;
}

forall(i in Is,t in T:t!=lastT)
{

ct12[i,t]:Vmax[i]*Nbar[i,t,t+1]>=sum(r in RIs)((abc[r,i])*(Rrt[r,t]));
}
forall(i in Is,t in T:t!=1)
{

ct14[i,t]:Vmax[i]*Nbar[i,t-1,t]>=sum(r in RIs)((abc[r,i])*(Rrt[r,t]));
}


  forall(r in R,t in T)
{
ct15[r,t]:Rmin[r]<=Rrt[r,t];
ct16[r,t]:Rmax[r]>=Rrt[r,t];
}
}

tuple NbarTuple {
int I1;
int T1;
int T2;
int value;}
{NbarTuple}NbarSet={<i,j,k,Nbar[i,j,k]>| i in I,j in T,k in T};


string outputRange; // = "ScheduledTrips!A:L";
int numRows; // = Shipments.size;

execute {
  numRows = NbarSet.size;
  outputRange = "cplex1.2!A2:" + "A" + (numRows+1);
  writeln(outputRange);
}
```

.dat file:

```
/*********************************************
 * OPL 12.6.1.0 Data
 * Author: Karthik Sundaram
 * Creation Date: 22 Jun 2016 at 10:52:28
 *********************************************/
SheetConnection File("cplex1.2.xlsx");

 zC=2;
 zI=34;
 zIco=27;
 zIemr=12;
 zR=33;
 zReq=8;
 zRint=18;
 zRpk=8;
 zT=8;
zIs=34;
zRIs=18;

 H from SheetRead (File, "Sheet1!B1");
 Cif from SheetRead (File, "Sheet1!B4:B30");
 Civ from SheetRead (File, "Sheet1!E4:E30");
 Cr from SheetRead (File, "Sheet1!H4:H36");
 Rmax from SheetRead (File, "Sheet1!K4:K36");
 Rmin from SheetRead (File, "Sheet1!N4:N36");
 Vmax from SheetRead (File, "Sheet1!B142:B175");
 Vmin from SheetRead (File, "Sheet1!E142:E175");
 abc from SheetRead (File, "Sheet1!W4:AC10");
a from SheetRead (File, "Sheet1!B39:B65");
b from SheetRead (File, "Sheet1!E39:E65");
delt from SheetRead (File, "Sheet1!E1");
deltmin from SheetRead (File, "Sheet1!H1");
lambda from SheetRead (File, "Sheet1!H39:AO71");
mu from SheetRead (File, "Sheet1!H73:AO105");
mubar from SheetRead (File, "Sheet1!H107:AO139");
nu from SheetRead (File, "Sheet1!H141:AH173");
nubar from SheetRead (File, "Sheet1!H175:AH207");
rhomax from SheetRead (File, "Sheet1!H209:H235");
rhomin from SheetRead (File, "Sheet1!K209:K235");
lastT from SheetRead (File, "Sheet1!K1");
Rro from SheetRead (File, "Sheet1!B71:B103");
NbarSet to SheetWrite(File,"Sheet6!A1:D2176");
```

8)**.mod file:**

```
/*********************************************
 * OPL 12.6.1.0 Model
 * Author: Karthik Sundaram
 * Creation Date: 8 Jul 2016 at 14:10:55
 *********************************************/
int zC=...;
int zI=...;
```

```
int zIco=...;
int zIemr=...;

int zR=...;

int zReq=...;
int zRint=...;
int zRpk=...;

int zT=...;

 range C = 1..zC;
  range I = 1..zI;


     range Ico = 1..zIco;
      range Iemr = 1..zIemr;
       range R = 1..zR;

         range Req = 1..zReq;
          range Rint = 12..zRint;
          range Rpk = 4..zRpk;
          range T = 1..zT;




//Parameters
int H=...;
float Cif[I]=...;
float Civ[I]=...;
float Cr[R]=...;
float Rmax[R]=...;
float Rmin[R]=...;
float Vmax[I]=...;
float Vmin[I]=...;
float a[I]=...;
float b[I]=...;
int delt=...;
float deltmin=...;
float lambda[R,I]=...;
float mu[R,I]=...;   //start
float mubar[R,I]=...; //end
float nu[R,I]=...;  //start
float nubar[R,I]=...;  //end
float rhomax[I]=...;
float rhomin[I]=...;
float Rro[R]=...;
int lastT=...;

//Decision variables
dvar boolean Nbar[I,T,T];
dvar float+ Rrt[R,T];
dvar float+ Tt[T];
```

```
dvar float+ zetabar[I,T,T];

constraint ct1[T,T,Ico];
constraint ct2[T,T,Ico];
constraint ct5[I,T];
constraint ct6[I,T];
constraint ct7[R,T];
constraint ct8[R,T];
constraint ct9[R,T];
constraint ct10[R,T];
constraint ct11[I,T];
constraint ct12[I,T];
constraint ct13[I,T];
constraint ct14[I,T];
constraint ct15[R,T];
constraint ct16[R,T];

dexpr float profit =(sum(r in R)(Cr[r]*(Rrt[r,lastT]))-
sum(i in I,t in T:t!=lastT,tp in T:t<tp&&tp<=t+delt)
  (Cif[i]*Nbar[i,t,tp]+Civ[i]*zetabar[i,t,tp]));


maximize (profit);

subject to
{
//Timing of tasks
//batch tasks

//zero wait tasks

//continuous tasks
forall(t,tp in T:tp==t+1&&t!=lastT,i in Ico)
{
ct1[t,tp,i]:Tt[tp]-Tt[t]>=zetabar[i,t,tp]/rhomax[i];
}
//continuous tasks that must exceed minimum processing rate
forall(t,tp in T:tp==t+1&&t!=lastT,i in Iemr :i in Ico)
{
ct2[t,tp,i]:Tt[tp]-Tt[t]<=H*(1-Nbar[i,t,tp])+zetabar[i,t,tp]/rhomin[i];
}

//Initialization
ct3:Tt[1]==0;
ct4:Tt[lastT]==H;

forall(i in Ico,t in T:t<lastT)
{
ct5[i,t]:deltmin*rhomin[i]*Nbar[i,t,t+1]<=zetabar[i,t,t+1];
ct6[i,t]:H*rhomax[i]*Nbar[i,t,t+1]>=zetabar[i,t,t+1];
}


forall(r in R, t in T)
{
if(t>1)
```

```
{
ct7[r,t]:Rrt[r,t]==Rrt[r,t-1]+sum(i in Ico:t<lastT)(mu[r,i]*Nbar[i,t,t+1]+
mubar[r,i]*Nbar[i,t-1,t]+lambda[r,i]*zetabar[i,t-1,t]);
}
else if (t==1)
{
ct8[r,t]:Rrt[r,t]==Rro[r]+sum(i in Ico:t<lastT)(mu[r,i]*Nbar[i,t,t+1]);
}
}


 forall (r in Rint, t in T)
  {
  ct9[r,t]:Rrt[r,t]==0;
  }
    forall(r in Rpk,t in T:t!=lastT)
   {
    ct10[r,t]:Rrt[r,t]==0;
}


  forall(r in R,t in T)
{
ct15[r,t]:Rmin[r]<=Rrt[r,t];
ct16[r,t]:Rmax[r]>=Rrt[r,t];
}
}
tuple NbarTuple {
int I1;
int T1;
int T2;
int value;}
{NbarTuple}NbarSet={<i,j,k,Nbar[i,j,k]>| i in I,j in T,k in T};


string outputRange; // = "ScheduledTrips!A:L";
int numRows; // = Shipments.size;

execute {
   numRows = NbarSet.size;
   outputRange = "cplex1.2!A2:" + "A" + (numRows+1);
   writeln(outputRange);
}
```

.dat file:

```
/*********************************************
 * OPL 12.6.1.0 Data
 * Author: Karthik Sundaram
 * Creation Date: 8 Jul 2016 at 14:11:08
 *********************************************/
SheetConnection File("cpl2.3.xlsx");

 zC=2;
 zI=27;
```

```
 zIco=27;
 zIemr=12;
 zR=33;
 zReq=8;
 zRint=18;
 zRpk=8;
 zT=8;

 H from SheetRead (File, "Sheet1!B1");
 Cif from SheetRead (File, "Sheet1!B4:B30");
 Civ from SheetRead (File, "Sheet1!E4:E30");
 Cr from SheetRead (File, "Sheet1!H4:H36");
 Rmax from SheetRead (File, "Sheet1!K4:K36");
 Rmin from SheetRead (File, "Sheet1!N4:N36");
 Vmax from SheetRead (File, "Sheet1!Q4:Q30");
 Vmin from SheetRead (File, "Sheet1!T4:T30");
a from SheetRead (File, "Sheet1!B39:B65");
b from SheetRead (File, "Sheet1!E39:E65");
delt from SheetRead (File, "Sheet1!E1");
deltmin from SheetRead (File, "Sheet1!H1");
lambda from SheetRead (File, "Sheet1!H39:AH71");
mu from SheetRead (File, "Sheet1!H73:AH105");
mubar from SheetRead (File, "Sheet1!H107:AH139");
nu from SheetRead (File, "Sheet1!H141:AH173");
nubar from SheetRead (File, "Sheet1!H175:AH207");
rhomax from SheetRead (File, "Sheet1!H209:H235");
rhomin from SheetRead (File, "Sheet1!K209:K235");
lastT from SheetRead (File, "Sheet1!K1");
Rro from SheetRead (File, "Sheet1!B71:B103");
NbarSet to SheetWrite(File,"Sheet4!A1:D1728");
```

9)**.mod file:**

```
/*********************************************
 * OPL 12.6.1.0 Model
 * Author: Karthik Sundaram
 * Creation Date: 18 Jul 2016 at 13:47:17
 *********************************************/

//Sets
int zC=...;
int zI=...;
int zIs=...;

int zIco=...;
int zIemr=...;
int zRIs=...;
int zR=...;
int zIb=...;
int zReq=...;
int zRint=...;
int zRpk=...;
int zRcl=...;
int zT=...;

 range C = 1..zC;
```

```
range Rcl=4..zRcl;
range Ib=35..zIb;
range I = 1..zI;
range RIs=12..zRIs;
range Is=28..zIs;
range Ico = 1..zIco;
range Iemr = 1..zIemr;
range R = 1..zR;

range Req = 1..zReq;
range Rint = 12..zRint;
range Rpk = 4..zRpk;
range T = 1..zT;



//Parameters
int H=...;
float Cif[Ico]=...;
float Civ[Ico]=...;
float Cr[R]=...;
float Rmax[R]=...;
float Rmin[R]=...;
float Vmax[I]=...;
float Vmin[I]=...;
float a[Ib]=...;
float b[Ib]=...;
int delt=...;
float deltmin=...;
float abc[RIs,Is]=...;
float xyz[C,I]=...;
float pqr[Rcl,Ib]=...;
float lambda[R,I]=...;
float mu[R,I]=...;   //start
float mubar[R,I]=...; //end
float nu[R,Ico]=...;  //start
float nubar[R,Ico]=...;  //end
float rhomax[Ico]=...;
float rhomin[Ico]=...;
float Rro[R]=...;
int lastT=...;

//Decision variables
dvar boolean Nbar[I,T,T];
dvar float+ Rrt[R,T];
dvar float+ Tt[T];
dvar float+ zetabar[I,T,T];

constraint ct1[T,T,Ico];
constraint ct2[T,T,Ico];
constraint ct5[I,T];
constraint ct6[I,T];
constraint ct7[R,T];
constraint ct8[R,T];
```

```
constraint ct10[R,T];

constraint ct12[I,T];

constraint ct14[I,T];
constraint ct15[R,T];
constraint ct16[R,T];

dexpr float profit =(sum(r in R)(Cr[r]*(Rrt[r,lastT]))-
sum(i in Ico,t in T:t!=lastT,tp in T:t<tp&&tp<=t+delt)
  (Cif[i]*Nbar[i,t,tp]+Civ[i]*zetabar[i,t,tp]));


maximize (profit);

subject to
{
//Timing of tasks
//batch tasks

//zero wait tasks

//continuous tasks
forall(t,tp in T:tp==t+1&&t!=lastT,i in Ico)
{
ct1[t,tp,i]:Tt[tp]-Tt[t]>=zetabar[i,t,tp]/rhomax[i];
}
//continuous tasks that must exceed minimum processing rate
forall(t,tp in T:tp==t+1&&t!=lastT,i in Iemr)
{
ct2[t,tp,i]:Tt[tp]-Tt[t]<=H*(1-Nbar[i,t,tp])+zetabar[i,t,tp]/rhomin[i];
}

//Initialization
ct3:Tt[1]==0;
ct4:Tt[lastT]==H;

forall(i in Ico,t in T:t<lastT)
{
ct5[i,t]:deltmin*rhomin[i]*Nbar[i,t,t+1]<=zetabar[i,t,t+1];
ct6[i,t]:H*rhomax[i]*Nbar[i,t,t+1]>=zetabar[i,t,t+1];
}


forall(r in R, t in T)
{
if(t>1)
{
ct7[r,t]:Rrt[r,t]==Rrt[r,t-1]+sum(i in Ico:t<lastT)(mu[r,i]*Nbar[i,t,t+1]+
mubar[r,i]*Nbar[i,t-1,t]+lambda[r,i]*zetabar[i,t-1,t])+
sum(i in Is:t<lastT)(mu[r,i]*Nbar[i,t,t+1]+mubar[r,i]*Nbar[i,t-1,t]);
}
else if (t==1)
{
ct8[r,t]:Rrt[r,t]==Rro[r]+sum(i in Ico:t<lastT)(mu[r,i]*Nbar[i,t,t+1]);
}
```

```
}

    forall(r in Rpk,t in T:t!=lastT)
   {
    ct10[r,t]:Rrt[r,t]==0;
}

forall(i in Is,t in T:t!=lastT)
{

ct12[i,t]:Vmax[i]*Nbar[i,t,t+1]>=sum(r in RIs)((abc[r,i])*(Rrt[r,t]));
}
forall(i in Is,t in T:t!=1)
{

ct14[i,t]:Vmax[i]*Nbar[i,t-1,t]>=sum(r in RIs)((abc[r,i])*(Rrt[r,t]));
}
//cleaning constraintss
forall(r in Rcl,t in T:t<lastT-1,c in C)
  {
 (sum(i in I)xyz[c,i]*mubar[r,i]*Nbar[i,t,t+1])+
 (sum(i in I)
 (xyz[1,i]+xyz[2,i])*(1-xyz[c,i])*mubar[r,i]*Nbar[i,t+1,t+2])<=1;
  }

forall(i in Ib,c in C,t in T:t>1 && t<lastT-2)
  {
sum(ibar in I)sum(r in Rcl)pqr[r,i]*xyz[c,ibar]*mubar[r,ibar]*Nbar[ibar,t-1,t]+
(sum(ibar in I)
sum(r in Rcl)(xyz[1,i]+xyz[2,i])*pqr[r,i]*(1-xyz[c,ibar])*mubar[r,ibar]
*Nbar[ibar,t+1,t+2])<=1+Nbar[i,t,t+1];
  }
//
  forall(r in R,t in T)
{
ct15[r,t]:Rmin[r]<=Rrt[r,t];
ct16[r,t]:Rmax[r]>=Rrt[r,t];
}

//forall(i in Ib)
 // {
 // sum(t in T:t>1 && t<lastT-2)Nbar[i,t,t+1]==1;
  //}

forall(i in Ib,t in T,tp in T:tp>t && t+delt>=tp && t!=lastT )
{
Vmin[i]*Nbar[i,t,tp]<=zetabar[i,t,tp];
Vmax[i]*Nbar[i,t,tp]>=zetabar[i,t,t+1];

}
 forall(i in Ib,t in T,tp in T:tp>t && tp<=t+delt &&t!=lastT)
   {
    Tt[tp]-Tt[t]>=a[i]*Nbar[i,t,tp];
   }
```

```
  forall(i in Ib,t in T,tp in T:tp>t && tp<=t+delt &&t!=lastT)
  {
   Tt[tp]-Tt[t]<=H*(1-Nbar[i,t,tp])+a[i]*Nbar[i,t,tp];
  }
}
```

```
tuple NbarTuple {
int I1;
int T1;
int T2;
int value;}
{NbarTuple}NbarSet={<i,j,k,Nbar[i,j,k]>| i in I,j in T,k in T};


string outputRange; // = "ScheduledTrips!A:L";
int numRows; // = Shipments.size;

execute {
  numRows = NbarSet.size;
  outputRange = "cplex1.2!A2:" + "A" + (numRows+1);
  writeln(outputRange);
}
```

.dat file

```
/*********************************************
 * OPL 12.6.1.0 Data
 * Author: Karthik Sundaram
 * Creation Date: 18 Jul 2016 at 13:48:21
 *********************************************/
SheetConnection File("cplex1.2new.xlsx");

 zC=2;
 zIb=38;
 zI=38;
 zIco=27;
 zRcl=7;
 zIemr=12;
 zR=33;
 zReq=8;
 zRint=18;
 zRpk=8;
 zT=8;
zIs=34;
zRIs=18;

 H from SheetRead (File, "Sheet1!B1");
 Cif from SheetRead (File, "Sheet1!B4:B30");
 Civ from SheetRead (File, "Sheet1!E4:E30");
 Cr from SheetRead (File, "Sheet1!H4:H36");
 Rmax from SheetRead (File, "Sheet1!K4:K36");
```

```
 Rmin from SheetRead (File, "Sheet1!N4:N36");
 Vmax from SheetRead (File, "Sheet1!B142:B179");
 Vmin from SheetRead (File, "Sheet1!E142:E179");
 abc from SheetRead (File, "Sheet1!W4:AC10");
 a from SheetRead (File, "Sheet1!B39:B42");
b from SheetRead (File, "Sheet1!E39:E42");
delt from SheetRead (File, "Sheet1!E1");
deltmin from SheetRead (File, "Sheet1!H1");
lambda from SheetRead (File, "Sheet1!H39:AS71");
mu from SheetRead (File, "Sheet1!H73:AS105");
mubar from SheetRead (File, "Sheet1!H107:AS139");
nu from SheetRead (File, "Sheet1!H141:AH173");
nubar from SheetRead (File, "Sheet1!H175:AH207");
rhomax from SheetRead (File, "Sheet1!H209:H235");
rhomin from SheetRead (File, "Sheet1!K209:K235");
lastT from SheetRead (File, "Sheet1!K1");
Rro from SheetRead (File, "Sheet1!B71:B103");

xyz from SheetRead (File, "Sheet1!B337:AM338");
pqr from SheetRead (File, "Sheet1!B341:E344");
NbarSet to SheetWrite(File,"Sheet11!A1:D3800");
```