Optimizing the planning of harvest, transport and grape crushing activities in the wine supply chain

1. Introduction

Supply Chain Management is important for wineries because they compete in an international marketplace where “old world wine producers” have suffered a decrease in wine production and in consumption in the last years, and where “new world wine producers” are becoming more aggressive, offering very high quality wines at more competitive prices. Therefore, improving supply chain’s effectiveness and efficiency becomes a critical factor to remain competitive and to develop competitive advantages.

In this paper, an approach to deal with the part of the wine supply chain that goes from the harvest to the grape crushing activities is presented.

The problem to be solved consists of daily scheduling in regards to harvest and loading, transporting in trucks, unloading and grape crushing activities in a winery, during the harvest season.

The winery works without breaks, from an opening time to a closing time. The farms have a morning schedule, then a siesta mid-day, and they continue working in the afternoon with a fixed schedule.

The main objective is to maximize the consumer demand. In order to fulfill this objective, the weight of grapes that will be crushed at the winery has to be maximized.

The problem has been solved with OPL, due to the possibility to use different optimization modules. The modules are Constraint Programming Module and CPLEX Module.

From the economic view point, the main objective is related to production, whereas the secondary objectives affect costs. Since sustainability [1] and reduction of gas emissions [2] are becoming more important as the time passes by, the optimization of the use of trucks and the consequently impact on the environmental preservation is a very relevant issue, especially in our province, where the trucks are in general old, and there is no gas emission control regulation.

2. Description of the problem

Different approaches have been researched in wine industry such as [3, 4, 5, 6]. For wine companies it is increasingly important to integrate logistics processes along the supply chain and to improve the performance of each process to reach a world-class standard.

This article deals with a small part of this Wine Supply Chain. Specifically, the first part of the chain that involves Grape Grower and Wine Producer.

The farms have blocks with different variety of grapes (malbec, syrah, chardonnay, etc) and with two possible conduction systems (Vertical Shoot Positioned Trellis and Horizontal High trellis). Although the term vineyard is more specific for grape, we use the term farm due to the fact that in some countries the term vineyard refers to a block. Therefore, the term farm will be used as a set of blocks.

The method used to load the truck can be traditional manual, assisted manual, or mechanical.

There are different kinds of trucks. Each type of truck determines the weight of grapes that it can carry.
In order to clarify the presentation, a particular example is used in the paper. For this example, there are three kinds of trucks: Dump Trucks, Ordinary Trucks, and Trucks with Trailer, that load 10, 15, and 30 tons of grapes respectively.

The schedule for the winery in the example has been set up to an opening time 9 AM and a closing time 9 PM. The morning schedule for the farms has been set up to 7:30 AM up to 12 AM, and for the afternoon schedule for the farms, 2 PM up to 7:30 PM.

As shown in the next figure, each truck has to perform four activities:

1. Harvest and loading
2. Transportation from farms to Winery
3. Unloading and grape crushing in Winery
4. Transportation from the winery to a farm with the Truck empty.

The duration of the crushing and unloading activities depends on the kind of truck.

We consider one winery and many farms and therefore the transport is simpler than other wine transportation problems such as [7]. The duration of the transportation activity depend only on the farm.

The farms have blocks with different variety of grapes (malbec, syrah, chardonnay, etc) and the method used to load the truck can be traditional manual, assisted manual, or mechanical.

The duration of the “Harvest and loading” activity [8] in our problem is a function of the Type of Truck, the Variety, the Method used to load the truck, and finally, if the method used to load the truck is manual (traditional or assisted), the quantity of workers.

The Type of Truck determines the quantity of tons to be loaded. The variety determines the harvest rate, that can take the values “low, medium or high”. Assuming that the method used to load the truck is traditional manual, the value “low” implies that one worker can harvest 50 kilos of grapes in one hour. Similarly, one worker can harvest 100 kilos of grapes in one hour for the value “medium” and 200 kilos for the value “high”.

The quantity of kilos to be loaded divided by the corresponding value (50, 100 or 200) give us the time (measured in hours) that would take one worker to harvest.

This time, times 60 give us the time in minutes and divided by the quantity of workers give us the duration of the activity “Harvest and loading” if the method used to load the truck is traditional manual.

If the method is “assisted manual”, the duration of the activity must be divided by 2.

Finally, if the method is mechanical, the duration does not require workers and the duration is the quantity of kilos of grapes divided by 1000, which is quantity of kilos per hour that can be harvested for the case of the mechanical method.
The main objective is to maximize the weight of grapes that will be crushed at the winery. The next figure helps to get a visual idea of the main objective. The objective function is described in the next section.

The secondary objectives are:
1. to reduce the wait time of the trucks and
2. to reduce the quantity of trucks.

The next figure shows the wait time of the trucks. The green double arrows represent the times that the trucks wait in the farms, whereas the purple color double arrows represent the times that the trucks wait in the winery.

The next figure shows the case in which the same truck can be used.
Finally, there are consistency constraint such as “if the conduction systems is Horizontal High trellis, the method to load the truck cannot be mechanical”.

3. Resolution of the Problem

Due to the fact that the whole program could be too long to explain, we are going to show the most relevant parts only.

The strategy to solve the problem is divided in the following steps:

1. To find the quantity of trucks of each type that maximizes the quantity of kilos that are going to be crushed in the winery.
2. To determine the quantity of activities in the winery and theirs durations.
3. To choose the order in which the activities in the winery are going to be sequenced and to schedule the rest of the activities.

We have used the IBM ILOG CPLEX Optimization Studio Version: 12.2, based on [9].

We use an integer programming model [10] to find the optimum quantity of trucks of each type to solve the first step. This is implemented in OPL by creating a Run Configuration with the CPLEX Optimizer.

The strategy to solve the steps 2 and 3 is implemented in the module that contains the script with the main control flow. In this module, we use the Constraint Programming Optimizer (using CP) only to define the activities and visualize them by means of a Gantt diagram, and to control the consistency of the solution.

The following figure shows the CP module and the CPLEX module. If Constraint Programming is used, it is necessary to specify it explicitly (var masterCp = cp).

If CPLEX is used, it is not necessary to specify it because OPL uses the CPLEX module by default.
The first step consists on creating a run configuration for the “unloading and grape crushing” activities, which is an object of the class `IloOplRunConfiguration`.

There are different kinds of trucks; each type of truck determines the weight of grapes that is carried.

The first step is to determine the quantity of trucks of each type that form the solution, in such a way that weight of grapes that will be crushed at the winery is maximized. To do that the CPLEX module is used.

The main part of the CPLEX module is the following.

```plaintext
maximize
  sum(truckType in TruckTypes)
  qTrucks[truckType] * qkg[truckType];

subject to {
  ct:
    sum(truckType in TruckTypes)
    qTrucks[truckType] * dur[truckType] <= TempWindowForWinery;
}
```
With range

\[ \text{TruckTypes} = 1..\text{quantOfTruckTypes}. \]

Where \text{quantOfTruckTypes} represents the quantity of types of trucks.

\( \text{qkg[truckType]} \) represents the quantity of kilos for a \text{truckType} type of truck.

The example in the pictures uses \text{quantOfTruckTypes} = 3.

\( \text{truckType} \) takes the values 1 (dump truck), 2 (normal truck) and 3 (truck with trailer), and finally \( \text{qTrucks[1]} = 10, \text{qTrucks[2]} = 15 \) and \( \text{qTrucks[3]} = 30. \)

\( \text{dur[truckType]} \) represents the duration of the “unloading and grape crushing” activity in the winery for a \text{truckType} type of truck, calculated as explained before.

\( \text{qTrucks[truckType]} \) is a domain variable.

\[
\sum (\text{truckType in TruckTypes}) \text{qTrucks[truckType]} \times \text{qkg[truckType]};
\]

represents the objective function to be maximized.

The final value of \( \text{qTrucks[truckType]} \) represents the quantity of trucks of type \text{truckType} for which the weight of grapes that will be crushed at the winery is maximum.

We use the \text{ct} constraint to ensure that the time necessary to perform the “unloading and grape crushing” activities in the winery fit in the temporal window in which the winery is open (\text{TempWindowForWinery}).

When this CPLEX module is solved, we already know the quantity of each type of trucks that fulfill the main objective. We have determined that \( \text{qTrucks[truckType]} \) trucks of type \text{truckType} will be used.

We integrate the outcome of the CPLEX module into a master module that will schedule the activities. For that purpose we use a run configuration that uses the CP engine.

Each truck has to perform the four activities mentioned earlier.

The first thing we have to determine is the order in which each crushing activity will be scheduled.

Then, we have to schedule the four activities for each truck.

Having found the scheduling for the crushing activities and the type of truck used, we can know the duration of the activity “Unloading and grape crushing in Winery” (\( \text{dur[truckType]} \)) and therefore we can schedule this activity. The first crushing activity will be scheduled at the time that the Winery opens. The following crushing activities will be scheduled sequentially, knowing that it will fit the Winery schedule because this constraint is imposed in the CPLEX module.

In order to schedule the “Harvest and loading” activity, we have to choose the farm from which the grape will be loaded. First, we have to exclude the farms that do not have enough quantity of kilos of grapes. The quantity of kilos of grapes required (determined by the type of truck) is \( \text{qkg[truckType]} \).
Given the start of the “Unloading and grape crushing in Winery” activity, and the quantity of kilos to be harvested, we have to select the farm that minimizes the wait time of the trucks. To do that, we first calculate the duration of the “Harvest and loading” activity for each not excluded farm.

The duration of the “Harvest and loading” activity depends on the quantity of kilos of grapes to be loaded ($q_{kg}[\text{truckType}]$), and also depends on the speed of the harvest and loading.

Let $\text{startActWinery}$ be the start of the “Unloading and grape crushing in Winery” activity, $\text{startBreak}$ and $\text{endBreak}$ be the times in which the break of the farms starts and ends.

For each not excluded farm, let $\text{transportTime}$ be the time corresponding to the transport from a farm to the winery and $\text{duration}$ the duration of the “Harvest and loading” activity.

Let us define

$$\text{startTransport} = \text{startActWinery} - \text{transportTime}$$

If $\text{startTransport} < \text{startBreak}$, then the staring time of the “Harvest and loading” activity will be

$$\text{startTransport} - \text{duration},$$

the truck can finish the “Harvest and loading” activity in the morning and there is no delay in the departure of the truck and therefore the farm will be selected without further analysis.

If $\text{endBreak} < \text{startTransport} - \text{duration}$, then the staring time of the “Harvest and loading” activity will be

$$\text{startTransport} - \text{duration},$$

the truck can perform the “Harvest and loading” activity after the break and there is no delay in the departure of the truck and therefore the farm will be selected as in the previous case, without further analysis.

Otherwise, the staring time of the “Harvest and loading” activity will be

$$\text{startBreak} - \text{duration}$$

and the delay will be

$$\text{startTransport} - \text{duration} - (\text{startBreak} - \text{duration}).$$

If there are farms with no delay, the algorithm will select the first farm found. Otherwise, the algorithm will select the farm with the minimum delay.

In this version of the algorithm, the strategy to solve the steps 2 and 3 is implemented in the module that contains the script with the main control flow. In this module, we use the Constraint Programming Optimizer (using CP) only to define the activities and visualize them by means of a Gantt diagram, and to control the consistency of the solution.

As shown in the next figure, the outcome of the CP module uses two structures to show the results. The Interval Variables (called Activity Variable in previous versions of OPL) and the State Function (State Resource in previous versions of OPL).
By clicking the icon associated to Interval Variables, a Gantt chart is displayed. As shown in the next figure, an Interval Variable represents one activity.
By clicking the icon associated to the State Functions, another Gantt chart is displayed. As shown in the next figure, a State Function represents all of the activities that are performed by the same truck.

It is important to clarify that the outcome of the Integer Programming Module (associated to the main objective) is the number of trucks of each type that maximize the weight of grapes that will be crushed at the winery. The next figure shows the values for the example used in the paper.

In the Schedule Module (associated to the secondary objectives), the order in which the activities are scheduled is determined.
4. Conclusions and Future work

In this paper a problem in the wine industry for local case was presented.

A simple model using Integer Programming and Scheduling was explained and solved suing OPL.

The contributions of this research include the definition and representation of a model for a small part of wine supply chain.

A set of guidelines were described as part of the case study for the purpose of further studies of similar problems.

This work intends to improve the efficiency and sustainability of the processes involved in our local wine industry.

This is the beginning of a research on a particular problem that we have in wineries in San Argentina.

As future work, we still have to exhaustively test the program and adjust it to accurately represent the real case and a comparison with similar problems in different countries should be considered.

5. Acknowledgments

This research was jointly developed with ….the School of Engineering (UNCuyo, Argentina) and École Nationale Supérieure de Génie de Systèmes Industriels (INPL, France), under the PREMER F-599 and ARFITEC ARF-08-04 programs.

We would like to thank Professor John Bartholdi and our colleagues from the Wine Supply Chain Council for their comments and discussions.

6. References


