Optimization applied to work assignment in flower crops

Sebastian Mesa[1], Juan Carlos Rivera[2], and Juan Felipe Torres[3]

[1]Mathematics Engineering, Universidad EAFIT, smesa@eafit.edu.co
[2]School of Science, Universidad EAFIT, jrivera6@eafit.edu.co
[3]Flores El Trigal, juan.felipetg@gmail.com

May 2016

1 Introduction

The floricultural industry is a very important part of the Colombian Economy. As stated by [Pro16], Colombia is the largest supplier of flowers of the region and the second worldwide and its flowers are exported mainly to the U.S., Canada, United Kingdom, and Russia. This industry has over 40 years of tradition and produces more than 1.500 flower species and along with Netherlands, Ecuador and Kenya, it supplies worldwide Demand.

Flores El Trigal is a company in the flower production industry. This company manages several processes related to the cultivation of the flower varieties that it produces. The company is divided into two units that manage different stages of the production: the propagation and the production unit. The production unit manages all the processes that start from receiving plant cuttings that will become flowers up to the harvest, storage, and shipping of the flowers generated from the cuttings, while the propagation unit manages all the processes that are involved in the production of the cuttings that will later become flower plants.

For both the production and the propagation units, the crop area is located in specific areas of the farms denominated blocks which in turn are divided into smaller parts called beds. Each block is covered by a greenhouse structure put in place to protect the plants from the weather. Each block has two rows of beds but the number of beds in each row and block varies depending on the size of the block. The beds are the zones in which the plants are actually planted. The need for this structure arises from the use of manual labor to care for plants as the workers need the roads created by the separation of the beds to walk.

In the propagation unit, the plant cuttings produced are generated from plants used specifically for this purpose denominated mother plants. These plants start out the same as the ones used for producing flower plants but are given a special treatment when they are very young so they don't form flowers but instead focus on the forming foliage from which the cuttings are taken. These plants start producing cuttings four weeks after being planted and keep doing so until a designated age in which some of the cuttings they produce stop meeting the quality standards required. During the time in the crops, each plant must be treated regularly with several products with different functions such as nurturing, pest, and disease control.

The harvesting process is done manually by a team of workers every day. The workers in charge of the task move around all the blocks in the farm looking for the plants that need harvesting. The plants have an estimated production of cuttings depending on how many days have passed since the last harvesting. The time taken to process each plant also varies according to those days. At the moment, the assignment of work is done manually trying to minimize the movement of the workers as much as possible so they can spend more time harvesting. Since there are certain products that need to be applied to the plants and these products can be harmful for humans, entrance to a block where one of the products has been applied is restricted for a range of time after the product application and this time depends on the product.

After being harvested, the cuttings are taken to a cold room in which they are stored until they are needed to fulfill an order. Since the cuttings are organic matter and quality standard needs to be kept, they can only remain in the cold room for a certain period of time before they can no longer be used for orders. The orders are dispatched on specific days for each client.

This work Proposes a mathematical programming model to support planning decisions centered around the harvest of flowers minimizing the amount of resources used and the waste generated from the harvest. The constraints of the model seek to fulfill the demands of flowers, respect workers' schedules and performance, availability of the flowers, and productivity of the flower plants according to it's time in the cycle of harvest.

2 Problem Statement

This work deals about waste reduction in flowers crops by using optimization techniques. Waste from the process of harvest may come from several sources. The first and most evident is the one generated from the cuttings in the cold room that are not used to fulfill orders before the time they stop being of use due to quality standards. A second one comes from the cuttings that are wasted when harvested from the plant after their time of use; as some fruits can become to ripe while on the tree, the cuttings can become too old while on the plant. The third one comes from the time the workers travel from one place to the other as they could be harvesting plants. The fourth and last one comes from time wasted in harvesting associated with the performance of the workers and the time in the cycle of harvest of the plants. This waste generates unnecessary costs for the company.

Fulfillment of the orders is a priority for the company. Each client's orders must be send in specific days of the week that depend on the client. Since orders from one week are used one week later, quantities not met for orders must be sent the next week within specific days for each client so they must consider on schedule.

On a real case, workers need to be supervised in their labor throughout the day; which arises the need of the creation of worker groups whose number is limited to the amount of supervisor available. Also, while workers generally work alone on a single bed, the need for cooperation may arise which arises the need for groups of beds.

The problem consists of assigning workers the beds each one needs to harvest every day minimizing the total waste and also the non fulfillment of the orders.. This problem is comprised of decisions on three different levels: the scheduling of the beds for harvest, the assignment of workers, and the routing. The problem is to be approached by the use of a mathematical programming model.

3 Previous Research

Optimization techniques have been widely used in the field of agriculture to solve several kind of problems coming from logistic processes, worker assignments, financial decisions, maintenance, and other areas.

Routing problems arise frequently on the industry when crops are involved given that, usually machines, need to move around the crop for harvesting purposes. For instance, [SN16] propose a multi-objective particle optimization approach applied to a sugarcane field using harvesting machines. In this case, the considerations involve the minimization of distances and the maximization of harvest yield. [FAS02] compares two heuristic algorithms, a simulated annealing and a genetic algorithm, used in a similar case for the planning of motion of agricultural vehicles.

Other very important application of optimization techniques in the field arises from general production planning purposes having in account different variables depending on the specific case and needs. [MM14] study a case in a citrus company using mathematical programming and robust optimization to account for uncertainty considering maturity curves for the oranges, storage, and their utilization. [APS15] discuses a more general case involving a lineal program formulation solved with a Branch and Price and Cut algorithm considering variables of land usage and crop yields and demands. Finally, [GSA15] study a case of optimization applied to planning in apple orchards using a mathematical programming model accounting for fruit ripeness and quality, minimizing the waste of fruit outside the quality standards and the labor costs.

4 Justification

Manual labor is preferred in most of the processes regarding the plants due to the care the people involved in the task and this care cannot be reproduced by machines. This is specially true for the harvesting of cuttings from the plants because of the care needed to correctly separate the cutting from the plants and the knowledge needed to know which cuttings are ready to be harvested.

The use of this kind of labor, having in account that a considerable amount of people is needed, implies that a big part of the production costs from the cuttings comes from worker salaries. Thus, organizing these people optimally so they can harvest what is needed generating as little waste as possible and fulfilling as much of the orders as possible should reduce the expenses of the company.

5 Model

The main detail of the model will be explained in this section. The full model can be found in the appendix. The model described does not account for the waste produced by the cutting in the storage room. Due to the complexity of the model proposed, the beds that need to be harvested will be estimated by the model but not the route that needs to be followed to harvest those beds. This routes will be estimated by a simple traveling salesman problem.

5.1 Objective Function

The objective function of the model is in gathers the calculations of the different kinds of waste generated by the bed harvested each day and how well were the orders for a given day met thus for it to be optimal it needs to be minimized.

$$Min: \sum_{d \in days} \sum_{v \in vars} UnF_{v,d} + sW_{v,d} + hW_{v,d} + timeW_d + transW_d$$

The variable $UnF_{v,d}$ gathers the amount of cuttings unfulfilled in orders for every day and variety of plant. $sW_{v,d}$ gathers the waste generated from storage.Since the dynamics of the storage are not modeled, this model assumes there is no cold room and the cuttings harvested each day that are not used generate this waste. $hW_{v,d}$ gathers the waste in terms of cuttings having in account the productivity of the plants according to the bed's time in the cycle of harvest. $timeW_d$ gathers the waste in terms of cuttings having in account the performance of the workers in a bed. $transW_d$ gathers a measure in terms of cuttings of the distance between the beds harvested each day for every group of workers.

5.2 Harvest

$$Harv_{v,d} = \sum_{bd \in beds} isH_{bd,d} * Q_{bd,d,v} * CPPf(d, v, age)$$

The harvest for a given variety on a given day designated by the variable $Harv_{v,d}$ is calculated by the sum of the of the amount of plants of said variety in each bed harvested multiplied by a factor that gives the productivity of the bed in terms of cuttings per plant depending on where in the cycle of harvest the bed is and how old.

Here, $isH_{db,d}$ indicates weather the bed bd was harvested on day d, $Q_{bd,d,v}$ indicates the amount of plants of variety v on day d on bed bd and $CPPf(d, v, age)$ is a function that indicates a rate of cuttings per plant depending where in the cycle of harvest is the bed. The amount calculated for harvest along with the parameter of the amount ordered for each day yield the values for the $UnF_{v,d}$ and $sW_{v,d}$ variables.

5.3 Waste from Harvest

$$hW_{v,d} = \sum_{bd \in beds} isH_{bd,d} * Q_{bd,d,v} * CPPDf(d, v, age)$$

This waste is calculated by comparing the harvest done to the best harvest day to harvest in terms of cuttings per plant per day. This restriction uses variables describes before but the function that calculates the factor to acquire a quantity in terms of cuttings differs from the one described before. Here, $CPPDf(d, v, age)$ is a function that calculates a factor based on the difference between the maximum rate of cuttings per plant per day and the rate of the day at which the bed is harvested and the amount of days that the bed has not been harvested.

5.4 Waste from Time Used

Time taken in harvesting differs depending on the frequency of harvest since searching for cuttings of the right maturity can be harder if the bed is harvested very frequently or if it has been to long since the last harvest. This restriction has a similar for to the ones described before but in this case the factor calculated is having in account a rate of cuttings per hour harvested for each day not harvested and the maximum rate that can be achieved on a given week.

5.5 Waste from transit

$$tansW_d = \sum_{bd1 \in beds} \sum_{bd2 \in beds} isH_{bd1,d} * isH_{bd2,d} * BDf(bd1, bd2)$$

This waste is calculated by by adding a measure in terms of cuttings of the distance between the beds that are harvested on a given day d.

5.6 Cycle of Harvest

The place or time of a given bed in the cycle of harvest is modeled having in account whether the bed was harvested the day before and in which place of the cycle was the bed on the day. This yields a series of logical constraints that can be modeled as numerical constraints. for a bed to be on the first day of the cycle on a given day, i.e. the bed was harvested the day before, a constraint such as $daysNoHarvest_{1,bd,d} = isHbd, d-1$ is put in place. for the rest of the days of the cycle the constraint in logical form is given by $daysNoHarvest_{i,bd,d} = \neg isHbd, d-1 \land daysNoHarvest_{i-1,db,d-1} \forall i \in HarvestDays$.
The plants can only have several days without being harvested before all the cuttings are lost because of their age since they stop meeting the quality standards needed. When this happens a process is done to the plants that resets them to the state of the first day of the cycle, thus the constraint of the first day is modified to include this reset. this can be seen in the complete model.

6 Work in Progress

testing with small cases is being done to the model to verify and validate its behavior regarding the fulfillment of the constraints modeled and the optimality of the choices taken. Tests consists of runs with controlled scenarios to observe if a desired behavior is actually being taken.

7 Future Work

After a full validation of the model, follows testing with real case scenarios and the implementation of the of the cold room dynamics that so far have not been accounted for. Since the size of the problem for a real case is considerable, the testing done in this phase also focuses on computational time needed to run the model.

8 Conclusions

Mathematical programming models are a powerful tool to model and optimize, they have considerable drawbacks when the relations between the variables on the model are not lineal, as happens in many cases, and when the problem the model is trying to solve has a considerable size. Although for some cases linealization solves in a way the problem of modeling the non lineal relationships between variables, it comes with the cost of adding more constraints and variables to the model; which, in problem of considerable size, can become an issue.

References

[APS15] Laurent Alfandari, Agnes Plateau, and Xavier Schepler. “Sustainable Optimization of Agricultural Production”. In: *European Journal of Operational Research* 241 (2015), pp. 872–879 (cit. on p. 4).

[FAS02] K.P. Ferentinos, K.G. Arvanitis, and N. Sigrimis. “Heuristic optimization Methods for motion Planning of autonomous agricultural vehicles”. In: *Journal of Global Optimization* 23 (2002), pp. 155–170 (cit. on p. 3).

[GSA15] Marcela C. Gonzalez-Araya, Wladimir E. Soto-Silva, and Luis G. Acosta Espejo. “Harvest Planning in apple Orchards Using an optimization Model”. In: *Handbook of Operations Research in Agriculture and the Agri-Food Industry* 224 (2015), pp. 79–105 (cit. on p. 4).

[MM14] Jose R. Munhoz and Reinaldo Morabito. “Optimization approaches to support decision making in the production planning of a citrus company: A Brazilian case study”. In: *Computers and Electronics in Agriculture* 107 (2014), pp. 45–57 (cit. on p. 3).

[Pro16] ProColombia. *Colombia’s flower industry keeps thriving for Mother’s Day celebration.* 2016. URL: `http://www.prnewswire.com/news-releases/colombias-flower-industry-keeps-thriving-for-mothers-day-celebration-300264506.html` (visited on 05/15/2016) (cit. on p. 1).

[SN16] Kanchana Sethanan and Woraya Neungmatcha. “Multi-Objective ParticleSwarm Optimization for Mechanical Harvester Route PLanning of SugarcaneField Operations”. In: *European Journal of Operational Research* (2016). DOI: `10.1016/j.ejor.2016.01.043` (cit. on p. 3).

A Full Mathematical Programming Model coded in AMPL

```
# maximum number of beds in a block
param bedNum;

# total number of blocks
param blockNum;

# total number of workers
param workerNum;

# total number of plant varieties
param varNum;

# number of weeks for the time window of the model
param weekNum;

# number of days in the harvest cycle
param harvestDaysNum;

# number of groups to cluster beds bed on each block
param blockGroupNum;

# number of workers groups desired
param workerGroupNum;

# day of the first week that the model will start the planning
# 1: Monday, 2: Tuesday, 3: Wednesday,
# 4: Thursday, 5: Friday, 6: Saturday, 7: Sunday
param startDay;

# set of workers defined by the number of workers provided
set workers := 1 .. (workerNum);

# set of block groups defined by the number of groups by
# blocks provided
set bGroups := 1 .. (blockGroupNum);

# set of workers groups defined by the number of workers provided
set wGroups := 1 .. (workerGroupNum);

# set of beds defined by the number of beds provided
set beds := 1 .. (bedNum);
```

```
# set of blocks defined by the number of blocks provided
set blocks := 1 .. (blockNum);

# set of week defined defined by the number of weeks that the model
# will span provided
set weeks := 1 .. (weekNum) ordered;

# set of days defined by the 7 days of the weeks
set days := 1 .. 7 ordered;

# set of varieties defined by the number provided
set vars := 1 .. (varNum);

# set of harvest days defined by the number provided
set harvestDays := 1 .. (harvestDaysNum) ordered;

# set representing the two parts of the shift
# each day: Morning and afternoon
set dayPart := {1,2};

# set of tuples for the actual beds on every block
set bb within {blocks,beds};

# set of tuples for the actual days that the model spans
set wd within {days,weeks};

# parameter that indicates a measure in cuttings for
# the distance between 2 blocks
param blockTravel {b1 in blocks, b2 in blocks};

# parameter that indicates the amount of work time for each
# part of the day on each day of the week
param workHours {d in days, dp in dayPart};

# parameter that indicates the amount of pants of a given variety
# on a given bed on a given week
param plantQuant {(bl, bd) in bb, v in vars, w in weeks} default 0;

# parameter that indicates how close in days is a bed to
# it's flash season
# a number >0 indicates that it is on the flash season
param daysToFlash    {(bl, bd) in bb} default -6;

# parameter that indicates the days from flash left to
# start regular cycle
param daysFromFlash  {(bl, bd) in bb} default 0;
```

```
# parameter that indicates the age of the bed at the day that
# the model starts
param startWeek   {(bl, bd) in bb};

# productivity parameter for every variety on all weeks
param rendEPP  {v in vars, aw in activeWeeks, hd in harvestDays}
default 0;

# standard performance parameter for every variaety on all weeks
param rendEPH  {v in vars, aw in activeWeeks, hd in harvestDays}
default 1;

# productivity parameter calculation for harvest waste calculation
param rendEPPD {v in vars, aw in activeWeeks, hd in harvestDays}
               = rendEPP[v,aw,hd]/hd;

# productivity in flash for evey variety
param flashRendEPP{v in vars};

# performance in flash for evey variety
param flashRendEPH{v in vars};

# maximum productivity on a given week for every variety for waste
# calculation
param maxrendEPPD{v in vars, aw in activeWeeks}
= max{hd in harvestDays}(rendEPPD[v,aw,hd]);

# maximum performance on a given week for every variety for waste
# calclulation
param maxrendEPH{v in vars, aw in activeWeeks}
= max{hd in harvestDays}(rendEPH[v,aw,hd]);

# parameter for unit conversion for waste calculations given time
# the units are Hours per Plant
param minHPP{v in vars, aw in activeWeeks}
= min{hd in harvestDays}(rendEPP[v,aw,hd]/rendEPH[v,aw,hd]);

# initial states of the beds in the cycle of harvest
param daysNH0{(bl, bd) in bb, hd in harvestDays} default 0;

# factor for every workers to get a worker's performance factor
param workerFactor {w in workers};

# amount ordered for a given variety ion a given day
param requestQuant {v in vars, d in days, w in weeks};
```

```
# variable that indicates the amount of cuttings
# harvested of a given variety on a given day
var harvest {v in vars, (d,w) in wd} >=0;

# variable that indicates the amount of waste generated from harvest
var waste   {v in vars, (d,w) in wd} >=0;

# variable that indicated te amount missing from orders generated from
# harvest
var missing {v in vars, (d,w) in wd} >=0;

# variables that gather the different kinds of waste fow a given day
var wasteDesp    {(d,w) in wd} >=0; # waste from transit
var wasteHarvest {(d,w) in wd} >=0; # waste from harvest
var wasteTime    {(d,w) in wd} >=0; # waste from time taken

# binary variable that indicates if a worker is on a workers group
# on a given day
var workerInGroup {wo in workers, wg in wGroups, (d,w) in wd} binary;

# binary variable that indicates if a bed is on a bed group
# on a given day
var bedInGroup {(bl, bd) in bb, g in bGroups, (d,w) in wd} binary;

# binary variable that indicates if a bed is harvested
# by a group of workers on a given day
var isH {(bl, bd) in bb, wg in wGroups, dp in dayPart, (d,w) in wd}
binary;

# variable that indicates if a group of beds is harvested
# on a given day
var isHG{g in bGroups, bl in blocks, wg in wGroups, dp in dayPart,
          (d,w) in wd} binary;

# variables that indicates the state of the the bed in the cycle of
# harvest for each day
var daysNH {(bl, bd) in bb, hd in harvestDays, (d,w) in wd} binary;
var befF   {(bl, bd) in bb, (d,w) in wd} binary;
var inF    {(bl, bd) in bb, (d,w) in wd} binary;
var inF1   {(bl, bd) in bb, (d,w) in wd} binary;
var inF2   {(bl, bd) in bb, (d,w) in wd} binary;
var afF    {(bl, bd) in bb, (d,w) in wd} binary;
var afF1   {(bl, bd) in bb, (d,w) in wd} binary;
var afF2   {(bl, bd) in bb, (d,w) in wd} binary;
```

```
#variables that represent the binary product of the two variables
# in the name
var isHxdaysNH {(bl,bd) in bb, wg in wGroups,
hd in harvestDays, dp in dayPart, (d,w) in wd} >=0;
var isHxbefF {(bl,bd) in bb, wg in wGroups, dp in dayPart,
          (d,w) in wd}>=0;
var isHxinF {(bl,bd) in bb,wg in wGroups, dp in dayPart,
          (d,w) in wd}>=0;
var isHxafF {(bl,bd) in bb, wg in wGroups, dp in dayPart,
          (d,w) in wd}>=0;
var isHGxBedInGroup {(bl, bd) in bb, g in bGroups,
eg in wGroups, dp in dayPart, (d,w) in wd} >=0;
var isHGxisHG{g1 in bGroups, b1 in blocks, g2 in bGroups,
b2 in blocks,wg in wGroups, dp in dayPart,(d,w) in wd } binary;

# variables that indicates the time used by each group on
# each part of the day
var time{wg in wGroups, dp in dayPart, (d,w) in wd}>=0;


# Objective function
minimize f:

sum{v in vars, (d,w) in wd}
    (waste[v,d,w] + missing[v,d,w] +
    wasteDesp[d,w] + wasteHarvest[d,w] + wasteTime[d,w]);


# constraint regarding waste from storage
subject to wasteRes {v in vars, (d,w) in wd}:

waste[v,d,w] >= harvest[v,d,w] - requestQuant[v,d,w];

#constraint regarding missing amounts from orders
subject to missingRes {v in vars, (d,w) in wd}:

missing[v,d,w] >= requestQuant[v,d,w] - harvest[v,d,w];


# constraint regarding calculation of harvested cuttings
subject to calcHarvest {v in vars, (d,w) in wd}:

harvest[v,d,w] = sum{wg in wGroups, (bl, bd) in bb, dp in dayPart}
( plantQuant[bl,bd,v,w]*
(
sum{hd in harvestDays}
```

```
                     (isHxdaysNH[bl,bd,wg,hd,dp,d,w]
                      *rendEPP[v,startWeek[bl,bd] + w,hd])
  + isHxbefF[bl,bd,wg,dp,d,w]*0
  + isHxinF[bl,bd,wg,dp,d,w]*flashRendEPP[v]
  + isHxafF[bl,bd,wg,dp,d,w]*0
  )
 );

# constraint regarding calculation of waste from harvest
subject to calcWasteHarvest { (d,w) in wd}:

wasteHarvest[d,w] = sum{v in vars}
(
 sum{wg in wGroups, (bl, bd) in bb, dp in dayPart}
( plantQuant[bl,bd,v,w]*
(
sum{hd in harvestDays}
                        (isHxdaysNH[bl,bd,wg,hd,dp,d,w]*hd
                         *(maxrendEPPD[v,startWeek[bl,bd] + w]
                           - rendEPPD[v,startWeek[bl,bd] + w,hd]))
  + isHxbefF[bl,bd,wg,dp,d,w]*0
  + isHxinF[bl,bd,wg,dp,d,w]*0
  + isHxafF[bl,bd,wg,dp,d,w]*0
      )
  )
 );


# constraint regarding calculation fo time taken
subject to calcTime {wg in wGroups, dp in dayPart, (d,w) in wd}:

time[wg,dp,d,w]* sum{wo in workers}
    (workerInGroup[wo,wg,d,w]*workerFactor[wo]) >=
sum{wo in workers}(workerInGroup[wo,wg,d,w])*
  sum{v in vars}
(
 sum{(bl, bd) in bb}
( plantQuant[bl,bd,v,w]*
(
sum{hd in harvestDays}
                         (isHxdaysNH[bl,bd,wg,hd,dp,d,w]*
                         (rendEPP[v,startWeek[bl,bd] + w,hd]*
                         (1/rendEPH[v,startWeek[bl,bd] + w,hd])))
  + isHxbefF[bl,bd,wg,dp,d,w]*0
  + isHxinF[bl,bd,wg,dp,d,w]*flashRendEPP[v]*(1/flashRendEPH[v])
  + isHxafF[bl,bd,wg,dp,d,w]*0
```

```
       )
    )
 );

# constraint regarding time limits for each part of the day
# each day of the week
subject to timeRes{wg in wGroups,dp in dayPart, (d,w) in wd}:

workHours[d,dp] >= time[wg,dp,d,w];

# constraint regaring calculation of waste generated by performance
# or time taken
subject to calcTimeWaste { (d,w) in wd}:

wasteTime[d,w] = sum{v in vars}
(
 sum{wg in wGroups, (bl, bd) in bb, dp in dayPart}
( plantQuant[bl,bd,v,w]*
(
sum{hd in harvestDays}
                          (isHxdaysNH[bl,bd,wg,hd,dp,d,w]
                         *(rendEPP[v,startWeek[bl,bd] + w,hd]
                        * (1/rendEPH[v,startWeek[bl,bd] + w,hd])
                          - minHPP[v,startWeek[bl,bd] + w])*
                          maxrendEPH[v,startWeek[bl,bd] + w])
  + isHxbefF[bl,bd,wg,dp,d,w]*0
  + isHxinF[bl,bd,wg,dp,d,w]*0
  + isHxafF[bl,bd,wg,dp,d,w]*0
      )
  )
 );


# constraint regarding calculation of waste regarding displacements
subject to calDespWaste {(d,w) in wd}:

wasteDesp[d,w] = sum{wg in wGroups, dp in dayPart, g1 in bGroups,
     g2 in bGroups, b1 in blocks, b2 in blocks}
     (isHGxisHG[g1,b1,g2,b2,wg,dp,d,w]*blockTravel[b1,b2]);

# linealization of binary product constraint
subject to isHGxisHGres {(d,w) in wd,wg in wGroups, dp in dayPart,
g1 in bGroups, g2 in bGroups, b1 in blocks, b2 in blocks}:

isHGxisHG[g1,b1,g2,b2,wg,dp,d,w] >= isHG[g1,b1,wg,dp,d,w] +
         isHG[g2,b2,wg,dp,d,w] -1;
```

```
# constraint regarding modeling of time of the beds before
# the regular cycle
subject to befFres {(bl, bd) in bb, (d,w) in wd}:

befF[bl,bd,d,w]*10000 >= daysToFlash[bl,bd] -
           (d - startDay + (w-1)*7);

subject to inFres {(bl, bd) in bb, (d,w) in wd}:

inF[bl,bd,d,w] = inF1[bl,bd,d,w] - inF2[bl,bd,d,w];
    # day 28 to 33 -> 6

subject to inF1res {(bl, bd) in bb, (d,w) in wd}:

inF1[bl,bd,d,w]*10000 >= (d - startDay + (w-1)*7) -
    daysToFlash[bl, bd];

subject to inF2res {(bl, bd) in bb, (d,w) in wd}:

inF2[bl,bd,d,w]*10000 >= (d - startDay + (w-1)*7) -
    daysToFlash[bl, bd] - 6;

subject to afFres {(bl, bd) in bb, (d,w) in wd}:

afF[bl,bd,d,w] = inF2[bl,bd,d,w] - afF1[bl,bd,d,w];

subject to afF1res{(bl,bd) in bb, (d,w) in wd}:

afF1[bl, bd,d,w]*10000 >= (d - startDay + (w-1)*7) -
    (daysToFlash[bl, bd] + 6 + daysFromFlash[bl, bd]);


# constraints regarding modeling of shift in a beds place in the
# cycle of harvest

# constraints for the first day of the week differs since
# it has to reference a different week

subject to initDaysNH{(bl, bd) in bb, hd in harvestDays}:

daysNH[bl,bd,hd,startDay,1] = daysNH0[bl,bd,hd];

subject to weekBeginDaysNH1{(bl,bd) in bb, (d,w) in wd :
          w > 1 and d = 1}:
```

```
daysNH[bl,bd,1,1,w] <= sum{wg in wGroups, dp in dayPart}
    (isH[bl,bd,wg,dp,7, prev(w, weeks)])
 + (-afF1[bl,bd,7,prev(w,weeks)] + afF1[bl,bd,1,w] )
     + daysNH[bl,bd,last(harvestDays),7,prev(w,weeks)];

subject to weekBegin2DaysNH1 {(bl,bd) in bb, (d,w) in wd :
         w > 1 and d = 1}:

daysNH[bl,bd,1,1,w] >= sum{wg in wGroups, dp in dayPart}
    (isH[bl,bd,wg,dp,7, prev(w, weeks)]);

subject to weekBegin3daysNH1 {(bl,bd) in bb, (d,w) in wd :
         w > 1 and d = 1}:

daysNH[bl,bd,1,1,w] >= (-afF1[bl,bd,7,prev(w,weeks)] + afF1[bl,bd,1,w] );

subject to weekBegin4DaysNH1 {(bl,bd) in bb, (d,w) in wd :
          w > 1 and d = 1}:

daysNH[bl,bd,1,1,w] >= daysNH[bl,bd,last(harvestDays),7,prev(w,weeks)];


subject to restDaysNH1{(bl,bd) in bb, (d,w) in wd: d > 1 or
                   (d <> startDay and w = 1)}:

daysNH[bl,bd,1,d,w] <= sum{wg in wGroups, dp in dayPart}
                  (isH[bl,bd,wg,dp,prev(d,days),w])
     + (-afF1[bl,bd,prev(d,days),w] + afF1[bl,bd,d,w])
                  + daysNH[bl,bd,last(harvestDays),prev(d,days),w];

subject to rest2DaysNH1 {(bl,bd) in bb, (d,w) in wd: d > 1 or
                   (d <> startDay and w = 1)}:

daysNH[bl,bd,1,d,w] >= sum{wg in wGroups, dp in dayPart}
              (isH[bl,bd,wg,dp,prev(d,days),w]);

subject to rest3DaysNH1 {(bl,bd) in bb, (d,w) in wd: d > 1 or
                  (d <> startDay and w = 1)}:

daysNH[bl,bd,1,d,w] >= (-afF1[bl,bd,prev(d,days),w] + afF1[bl,bd,d,w]);

subject to rest4DaysNH1 {(bl,bd) in bb, (d,w) in wd: d > 1 or
                  (d <> startDay and w = 1)}:

daysNH[bl,bd,1,d,w] >= daysNH[bl,bd,last(harvestDays),prev(d,days),w];
```

```
subject to weekBeginDaysNH2P {(bl,bd) in bb,hd in harvestDays,
          (d,w) in wd : w > 1 and d = 1 and hd > 1 }:

daysNH[bl,bd,hd,1,w] >=
(1 - sum{wg in wGroups, dp in dayPart}(isH[bl,bd,wg,dp,7, prev(w,weeks)]))
+ daysNH[bl,bd,prev(hd,harvestDays),7,prev(w,weeks)] -1;

subject to restDaysNH2P {(bl,bd) in bb, hd in harvestDays, (d,w) in wd
     : hd > 1 and (d >1 or (d <> startDay and w = 1))}:

daysNH[bl,bd,hd,d,w] >=
    (1 - sum{wg in wGroups, dp in dayPart}(isH[bl,bd,wg,dp,prev(d,days),w]))
    + daysNH[bl,bd,prev(hd,harvestDays),prev(d,days),w] -1;

# constraint that indicate thtat a bed can only be
# at one place in the cycle of harvest on a given day
subject to oneStatePerDay {(bl,bd) in bb, (d,w) in wd}:

sum{hd in harvestDays}(daysNH[bl,bd,hd,d,w])
          + befF[bl,bd,d,w] + inF[bl,bd,d,w] + afF[bl,bd,d,w] =1;


# linealization of binary products
subject to isHxdaysNHres {(bl,bd) in bb, wg in wGroups, hd in harvestDays,
   dp in dayPart, (d,w) in wd}:

isHxdaysNH[bl,bd,wg,hd,dp,d,w] >= isH[bl,bd,wg,dp,d,w] +
         daysNH[bl,bd,hd,d,w] -1;

subject to isHxdaysNHres1 {(bl,bd) in bb, wg in wGroups, hd in harvestDays,
   dp in dayPart, (d,w) in wd}:

isHxdaysNH[bl,bd,wg,hd,dp,d,w] <= isH[bl,bd,wg,dp,d,w];

subject to isHxdaysNHres2 {(bl,bd) in bb, wg in wGroups, hd in harvestDays,
   dp in dayPart, (d,w) in wd}:

isHxdaysNH[bl,bd,wg,hd,dp,d,w] <= daysNH[bl,bd,hd,d,w];


subject to isHxbefFres {(bl,bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd}:

isHxbefF[bl,bd,wg,dp,d,w] >= isH[bl,bd,wg,dp,d,w]
              + befF[bl,bd,d,w] -1;
```

```
subject to isHxbefFres1 {(bl,bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd}:

isHxbefF[bl,bd,wg,dp,d,w] <= isH[bl,bd,wg,dp,d,w];

subject to isHxbefFres2 {(bl,bd) in bb, wg in wGroups, dp in dayPart,
           (d,w) in wd}:

isHxbefF[bl,bd,wg,dp,d,w] <= befF[bl,bd,d,w];


subject to isHxinfFres {(bl,bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd}:

isHxinF[bl,bd,wg,dp,d,w] >= isH[bl,bd,wg,dp,d,w] + inF[bl,bd,d,w] -1;

subject to isHxinFres1 {(bl,bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd}:

isHxinF[bl,bd,wg,dp,d,w] <= isH[bl,bd,wg,dp,d,w];

subject to isHxinFres2 {(bl,bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd}:

isHxinF[bl,bd,wg,dp,d,w] <= inF[bl,bd,d,w];


subject to isHxafFres {(bl,bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd}:

isHxafF[bl,bd,wg,dp,d,w] >= isH[bl,bd,wg,dp,d,w] +
              afF[bl,bd,d,w] -1;

subject to isHxaffFres1 {(bl,bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd}:

isHxafF[bl,bd,wg,dp,d,w] <= isH[bl,bd,wg,dp,d,w];

subject to isHxafFres2 {(bl,bd) in bb, wg in wGroups, dp in dayPart,
           (d,w) in wd}:

isHxafF[bl,bd,wg,dp,d,w] <= afF[bl,bd,d,w];


# constraint that indicates that a bed can't be harvested before the flash
```

```
subject to NoBefFHarvest {(bl,bd) in bb, wg in wGroups,dp in dayPart,
            (d,w) in wd}:

isHxbefF[bl,bd,wg,dp,d,w] =0;


# constraint that indicates that a bed cant be harvested after the flash
# before the start of the regular cycle
subject to NoAfFHarvest {(bl,bd) in bb, wg in wGroups,dp in dayPart,
            (d,w) in wd}:

isHxafF[bl,bd,wg,dp,d,w] = 0;


# constraint that indicates that a bed can only be harvested
# one time in the flash season
subject to OneFlashHArvest{(bl,bd) in bb: daysToFlash[bl,bd] > -6}:

sum{wg in wGroups, dp in dayPart, (d,w) in wd}
              (isHxinF[bl,bd,wg,dp,d,w]) = 1;

# constant that indicates whether a bed was harvested
# a bed is harvested of a group of workers harvests the group of beds
# to which the bed belongs
subject to isHarvestRes {(bl, bd) in bb, wg in wGroups, dp in dayPart,
          (d,w) in wd}:

isH[bl,bd,wg,dp,d,w] = sum{g in bGroups}
                (isHGxBedInGroup[bl,bd,g,wg,dp,d,w]);

# binary product linelization
subject to isHGxBedinG {(bl, bd) in bb, wg in wGroups, dp in dayPart,
  (d,w) in wd, g in bGroups}:

isHGxBedInGroup[bl,bd,g,wg,dp,d,w] >= isHG[g,bl,wg,dp,d,w]
               + bedInGroup[bl,bd,g, d,w] -1;

subject to isHGxBedinG1 {(bl, bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd, g in bGroups}:

isHGxBedInGroup[bl,bd,g,wg,dp,d,w] <= isHG[g,bl,wg,dp,d,w];

subject to isHGxBedinG2 {(bl, bd) in bb, wg in wGroups, dp in dayPart,
            (d,w) in wd, g in bGroups}:

isHGxBedInGroup[bl,bd,g,wg,dp,d,w] <= bedInGroup[bl,bd,g,d,w];
```

```
# constraint that indicate that a bed can onl be harvested
# one time on a day
subject to OneHarvestRes{(bl,bd) in bb, (d,w) in wd}:

sum{wg in wGroups, dp in dayPart}(isH[bl,bd,wg,dp,d,w]) <=1;

# constraint that indicates that a worker can only belong to
# one workers group on a day
subject to OneGroupPerWorker {wo in workers,(d,w) in wd}:

sum{wg in wGroups}(workerInGroup[wo,wg,d,w]) = 1;


# constraint that indicates that a bed can belong at most
# to one group of beds on a day
subject to OneGroupPerBed {(bl,bd) in bb, (d,w) in wd}:


sum{g in bGroups}(bedInGroup[bl,bd,g,d,w]) <=1;
```