



**HAL**  
open science

## Integration of Operational Constraints to Optimize Differential Harvest in Viticulture

Nicolas Briot, Christian Bessiere, Bruno Tisseyre, Philippe Vismara

► **To cite this version:**

Nicolas Briot, Christian Bessiere, Bruno Tisseyre, Philippe Vismara. Integration of Operational Constraints to Optimize Differential Harvest in Viticulture. ECPA: European Conference on Precision Agriculture, Jul 2015, Tel Aviv, Israel. pp.487-494, 10.3920/978-90-8686-814-8\_60 . lirmm-01275568

**HAL Id: lirmm-01275568**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01275568>**

Submitted on 18 Oct 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Integration of Operational Constraints to Optimize Differential Harvest in Viticulture.

N. Briot<sup>1</sup>, C. Bessiere<sup>1</sup>, B. Tisseyre<sup>2</sup> and P. Vismara<sup>1,3</sup>

<sup>1</sup>LIRMM, CNRS, Université de Montpellier, 161 rue Ada, 34095 Montpellier, France

<sup>2</sup>UMR ITAP, Montpellier SupAgro / Irstea, 2 Place Viala, 34060 Montpellier, France.

<sup>3</sup>MISTEA, Montpellier SupAgro, INRA, 2 Place Viala, 34060 Montpellier, France.

{briot, bessiere, vismara}@lirmm.fr

bruno.tisseyre@supagro.fr

## Abstract

One of the most important applications of precision agriculture to viticulture is the management of the quality of grapes at the field level through selective harvest. The recent developments of prototypes of conventional grape harvesting machines are able to sort two types of harvest quality. In this paper, we propose a formal description of the problem of differential harvesting. The objective is to optimize the routing of the grape harvester under several constraints. Our approach is based on Constraint Programming (CP), which is a powerful paradigm to solve combinatorial problems. We give a formulation of the problem as a Constraint Optimization Problem. The last section of the paper is devoted to preliminary experimental results on real data from a French vineyard.

**Key words:** precision viticulture, differential harvest, constraint programming.

## Introduction

One of the most important applications of precision agriculture applied to viticulture is managing grape quality at the within-field level through selective harvest (Tisseyre *et al.*, 2007). In viticulture, many studies have proposed defining within-field quality zones (Lamb *et al.*, 2004, Bramley and Hamilton, 2004). They demonstrated the technical and economic value of differential harvesting of these different zones. This interest justifies the recent development of prototypes of conventional grape harvesting machines able to sort two types of harvest quality as the EnoControl™ system prototype (newHolland Agriculture, PA, USA). These grape harvesting machines have two hoppers able to differentiate two types of grape quality (A and B) according to the harvested zone.

Assuming that the winery has the necessary organization to manage two qualities in parallel, the development of this new type of grape harvester raises new questions about the logistical organization of the harvest. The authors are aware of other differential harvesting organizations that involve the use of a grape harvester with side conveyor towards follower bins (Bramley and Hamilton, 2004). However, conventional grape harvesters are commonly used in Europe and this specific problem is particularly interesting to study. Optimizing harvest tasks requires minimizing the time of the operation and in particular the stops to empty hoppers. Ideally, this goal requires that both hoppers (A and B) of the machine are full at each stop. In the case of selective harvesting, the simultaneous filling of the two hoppers is difficult. Indeed, the hopper A will fill faster in the A quality zone (and vice versa for the hopper B). Other issues have

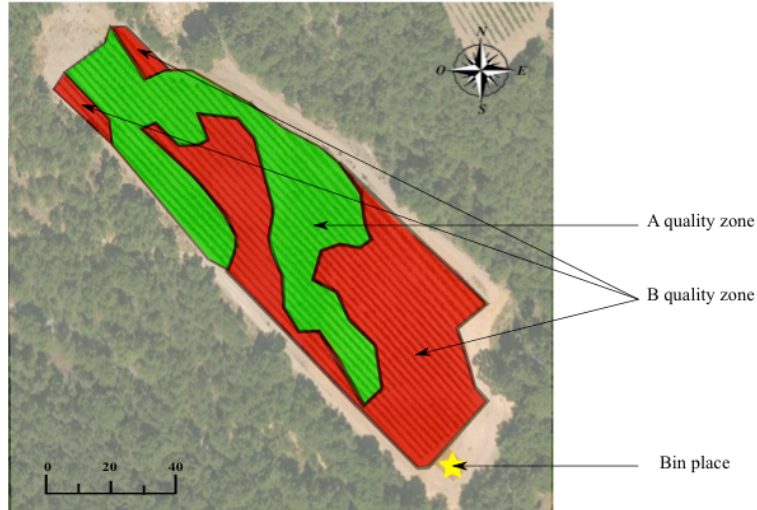
to be considered; for example: top quality grapes should not be altered (mixed) to lower quality grapes.

This paper proposes a formal description of the problem of differential harvesting with two hoppers and provides a tool to optimize the path of the harvesting machine. The problem is to optimize the grape harvester routing under several constraints. For instance, suppose that the objective is to collect a given amount of A quality grapes whereas all the other grapes will be considered as B quality ones. Then, some A quality grapes can be dumped into the hopper B, provided that the required amount of A quality grapes will finally be achieved. Many other constraints complicate the definition of the problem: the turning radius of the harvester, the position of the grape transporter (that can be fixed or follow the harvester on the side of the field), the shape of the vine field... All these constraints give to the problem an intrinsic combinatorial complexity. A preliminary approach to solve the issue of differential harvesting with two hoppers is presented here. It is based on Constraint Programming (CP), which is a powerful paradigm for solving combinatorial problems. To our knowledge, this is the first time CP is applied to viticulture. This paper gives a formulation of the problem as a Constraint Optimization Problem and presents preliminary implementation results with the CP solver Abscon (Merchez *et al.*, 2001).

## **Material and Methods**

### Field data

The approach was tested on a real data field from an experimental vineyard of INRA Pech-Rouge (Gruissan, Aude, France) located in southern France (N 43°08'4137, E 3°07'38.11, WGS84) (see **Erreur ! Source du renvoi introuvable.**). The field of 0.6 ha was established in 1990 with 1m spacing between vines and 2.5 m between rows. The vine field is made of 16 rows and the harvesting machine has a maximal capacity of 900 l for each hopper. Delineation of quality zones was based on vigor zones defined from airborne images acquired at veraison following methods outlined by Santesteban *et al.* (2013). Yield and level of quality of the different zones were estimated according to target sampling and references from the previous year. Note the estimation of the yield and the level of quality of early defined zones (several days before harvest) is a real issue in viticulture (Urretavizcaya *et al.*, 2013). This aspect was not under the scope of this study, therefore, yield and quality was assumed properly estimated over the zones of the field.



**Figure 1 Data: vine field with two qualities. Three B-quality zones (in red) and one A-quality zone (in green). The bucket is placed in the south of the field.**

### Differential Harvester Problem

The *Differential Harvest Problem* (DHP) occurs when grapes are mature. Harvesting must be done quickly to avoid losing quality and prevent bad weather damage. In this paper, we consider a vine field with two quality of grapes. The first one will be denoted *A-grapes* and corresponds to high quality grapes. These grapes will be used to produce high quality wine. The rest of the harvest will be labeled with the *B-grapes* class. Note that *A-grapes* can be downgraded to *B* class. But the converse is not true. *B-grapes* are used to produce less good quality wines like bulk wines, for example.

Conventional harvesting machines have two tanks called *hopper*. We note  $C_{max}$  the maximum capacity of one hopper. With such a machine, two categories of grapes can be separated. Generally, one hopper (called *a-hopper*) receives only *A-grapes* whereas the other one (called *b-hopper*) can contain *A-grapes*, *B-grapes* or both. It is important not to contaminate the grapes in *a-hopper* with *B-grapes*. When the hoppers are full, the machine can empty them into a bin located around the plot before continuing its work. We denote  $\delta$  the time required to empty the two hoppers in the bin.

The harvesting machine takes some time between picking grapes and putting them into hoppers. This time required to empty the conveyor belt is called *latency*. When the harvesting machine passes over two zones, the quality of grapes harvested is not guaranteed during this time because latency may lead to mixing of both quality grapes. Two scenarios are possible. First, when the machine leaves an *A-grapes* zone to enter a *B-grapes* zone, the grapes must be considered as *B-grapes*. The second case occurs when the machine passes from *B-grapes* to *A-grapes*. In this case, the grapes are also considered as *B-grapes* during the whole latency delay. Note that for the same row, the type and the number of transitions may vary according to the direction that the row is harvested. Thus, because of latency, the quantity of *A-grapes* (resp. *B-grapes*) that are collected within a given row can change with the direction. For instance, consider a row where the quality grapes areas sequence is *A-B-A-B* (see Figure 2). If the machine harvests the row from left to right, there is only one *B-A* transition. If the machine harvests the row in the opposite direction, two transitions *B-A* occur and therefore twice the risk of putting *B-grapes* into the *a-hopper* dedicated to *A-grapes*. Thus, quantities of *A-grapes* and *B-grapes* harvested will necessarily depend on the direction in which the

rows are harvested.

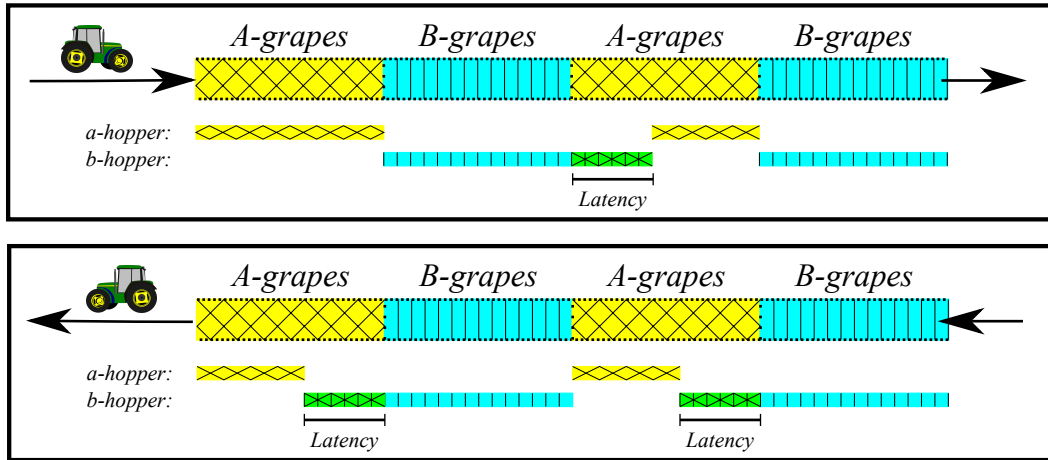


Figure 2. Quantities of grapes depend on the direction of row harvest.

$R_{min}$  is the desired minimum volume of *A-grapes* harvested in the vine field.  $R_{min}$  corresponds to a threshold. In practice,  $R_{min}$  corresponds to an *A-grapes* volume shortfall according to the objectives of the winery. If the vineyard contains more than  $R_{min}$  *A-grapes*, the excess amount will be downgraded to *B-grapes* quality. For this experiment,  $R_{min}$  is estimated at 1700 l. Until  $R_{min}$  has been reached, *A-grapes* are stored into the *a-hopper*. Once  $R_{min}$  has been reached and after the hoppers have been emptied, *A-grapes* and *B-grapes* can be mixed in both hoppers. Regarding  $R_{min}$ , there are three possibilities to fill the two hoppers. The first one occurs when the harvesting machine must differentiate both qualities in both hoppers (see Figure 3.a). When *a-hopper* is full, the machine can harvest *A-grapes* in the *b-hopper* (see Figure 3.b). In this case, these *A-grapes* are downgraded. Once  $R_{min}$  has been reached, the machine can mix grapes in both hoppers (see Figure 3.c).

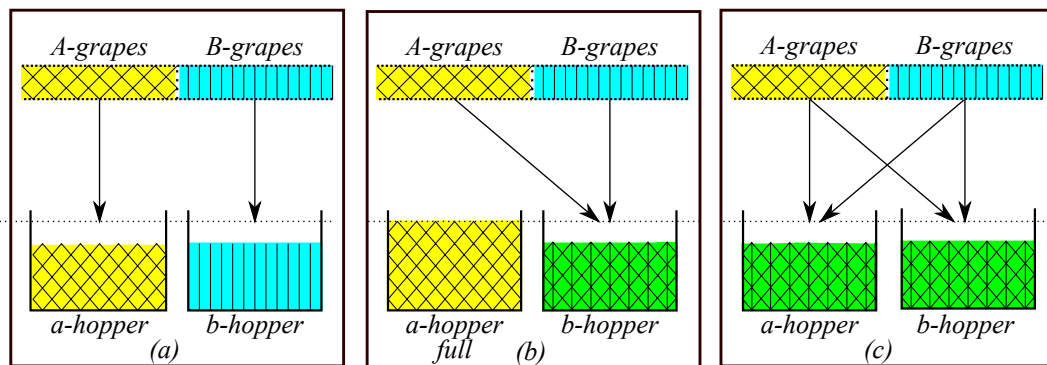


Figure 3. Three possibilities to fill the hoppers. (a) *A-grapes* and *B-grapes* are separated. (b) When *a-hopper* is full, *A-grapes* and *B-grapes* are mixed in the *b-hopper*. (c) Once  $R_{min}$  has been reached, *A-grapes* and *B-grapes* are mixed in the two hoppers.

The vine field was modeled with  $n$  rows by differentiating the two extremities of each row. A row  $i \in \{1, \dots, n\}$  is represented by extremities  $2i$  and  $2i+1$ . For each extremity, the quantities of *A-grapes* and *B-grapes* that will be collected have to be defined depending on the direction that the row is harvested. This quantity was computed according on the latency. For instance,  $Q^A(2i) = \alpha$  and  $Q^B(2i) = \beta$  on the  $2i$  extremity if

the row  $i = (2i, 2i+1)$  has  $\alpha$  *A-grapes* and  $\beta$  *B-grapes* in the  $2i \rightarrow 2i+1$  direction and  $Q^A(2i+1) = \gamma$  and  $Q^B(2i+1) = \sigma$  on the  $2i+1$  extremity if there are  $\gamma$  *A-grapes* and  $\sigma$  *B-grapes* in the  $2i+1 \rightarrow 2i$  direction.

Another important issue is the cost of the path between two extremities of different rows and between row extremities and the bin place. In the experimental field, the bin has been placed on the southern side. We denote  $d(p, q) = d(q, p) \forall p, q \in \{1, 2, \dots, 2n+1\}$  the time required to go from an extremity  $p$  to an extremity  $q$  (where 1 denote the bin place). This cost depends on the distance and turning radius of the harvesting machine.

**Definition (Differential Harvesting Problem):** Given a wine field described by a cost path matrix between row extremities (or the bin place) and an estimation of the quantity of *A-grapes* and *B-grapes* on each row according to the direction, given a harvesting machine with a hopper capacity of  $C_{max}$  and a latency to pass from B to A areas, given an objective of  $R_{min}$  *A-grapes* to harvest, find a sequence of extremities (that is, an order and orientation of the rows) that minimizes the time required to harvest the wine field and ensures at least  $R_{min}$  *A-grapes*.

### CSP formalism

We model the Differential Harvest Problem as a *Constraint Satisfaction Problem* (CSP). This formalism allows modeling discrete non-linear constraint optimization problems efficiently. A CSP involves a set of variables with domains of possible values and a set of constraints on these variables. A solution to the constraint network is an assignment of values to its variables that satisfy all its constraints. Constraints specify combinations of values that given subsets of variables are allowed to take. A constraint  $c$  is a relation defined on a sequence of variables  $X(c) = (x_{i_1}, \dots, x_{i_{|X(c)|}})$ , called the scheme of  $c$ .  $c$  is the subset of  $\mathbb{Z}^{|X(c)|}$  that contains the combinations of values (or tuples) that satisfy  $c$ .  $|X(c)|$  is called the arity of  $c$ . A constraint can be specified extensionally by the list of satisfying tuples, or intensionally by a Boolean function. A constraint network is composed of: a finite sequence of integer variables  $X = (x_1, \dots, x_n)$ , a domain for  $X$ , that is, a set  $D = D(x_1) \times \dots \times D(x_n)$ , where  $D(x_i) \subset \mathbb{Z}$  is the finite set of values that variable  $x_i$  can take, and a set of constraints  $C = \{c_1, \dots, c_e\}$ , where variables in  $X(c_j)$  are in  $X$ .

Constraint programming is made effective by the intensive use of constraint propagation. Constraint propagation reduces the domain of variables. For instance, consider three variables  $x, y, z$ , their domains  $D(x) = \{0, 2, 4\}$ ,  $D(y) = \{4, 6, 8\}$  and  $D(z) = \{1, 2, 3, 4, 5, 6, 7, 8\}$ , and the constraint  $x + y = z$ . In this problem, constraint propagation can reduce the domain of variable  $z$  because  $z$  can take only even values. It is also possible to remove value 2 from the domain of variable  $z$  because it cannot satisfy the constraint whatever the values of  $x$  and  $y$ . After constraint propagation, the domain of  $z$  becomes  $D(z) = \{4, 6, 8\}$ .

CSP solvers use backtrack search to find solutions. The search space is generally dramatically reduced thanks to constraint propagation or thanks to variable and value ordering heuristics. Variable (resp. value) ordering heuristics select variables (resp. values) in an order that is expected to reduce search.

### Differential Harvester Problem as a CSP

We give a model of the Differential Harvest Problem as a Constraint Satisfaction Problem.

*Variables and domains:*

- $\{row_1, \dots, row_{2n}\}$ : are the main variables. They denote the position of the harvesting machine in the vine field at each step.  $row_{2j-1}$  takes as value the extremity by which the harvester machine will enter in its  $j$ th harvested row.  $row_{2j}$  takes as value the extremity by which the machine goes out of the  $j$ th row. The domain for these variables is the set of possible extremities:  $D(row_i) = \{2, \dots, 2n + 1\}$ .
- $\{tub_1, \dots, tub_n\}$ : are binary variables. If  $tub_j = 1$ , the machine goes to the bin after the position  $row_{2j}$  and conversly, if  $tub_j = 0$ , the machine does not go to the bin after the position  $row_{2j}$ .
- $\{contract_1, \dots, contract_n\}$ : are binary variables linked to the objective of *Rmin A*-grapes to harvest. If the contract is achieved after  $row_{2j}$  then  $contract_j = 1$  else  $contract_j = 0$ .
- $\{U_1^a, \dots, U_n^a\}$  (resp.  $\{U_1^b, \dots, U_n^b\}$ ): are the variables that represent quantity of grapes in *a-hopper* (resp. *b-hopper*) at the position  $row_{2j}$  before a possible emptying. Their domain is  $D(U_1^a) = D(U_1^b) = \{0, \dots, 2 \times Cmax\}$ .
- $\{u_1^A, \dots, u_n^A\}$  (resp.  $\{u_1^B, \dots, u_n^B\}$ ): are the variables that represent quantity of *A-grapes* (resp. *B-grapes*) harvested in each rows.  $D(u_j^A) = D(u_j^B) = \mathbb{N}$ .
- $\{Ac_0, \dots, Ac_n\}$ : represent the total amount of grapes harvested from the beginning of the harvest.  $D(Ac_j) = \mathbb{N}$ .
- $\{\Delta_1, \dots, \Delta_{n-1}\}$ : represent the time from position  $row_{2j}$  to position  $row_{2j+1}$  with a possible passage to the bucket (if  $tub_j = 1$ ). The domain is  $D(\Delta_j) = \mathbb{N}$ .

*Constraints:*

First, the global constraint *alldifferent* requires that all variables are pairwise different. Here, this means that the harvesting machine passes one and only one time through each extremity.

$$Alldifferent(row_1, row_2, \dots, row_{2n}) \quad (1)$$

The second constraint implies that if the harvester machine enters in a row by extremity  $row_i$  then the next variable  $row_{i+1}$  must takes the other extremity of the same row.

$$|row_{2j-1} - row_{2j}| = 1 \quad \forall j \in \{1, \dots, n\} \quad (2)$$

Each extremity is associated with both quantities of grapes (A and B). The  $row_{2j-1}$  variable represents the extremity of row where the harvesting machine enters. Thus,  $u_j^A$  and  $u_j^B$  represents quantity of grapes in row when the harvesting machine enters in  $row_{2j-1}$  extremity. The quantity of grapes in row is a data of the problem. It can be stored as a constraint in extension: a set of allowed tuples. For example, if  $row_{2j-1} = 3$  then  $u_j^A = Q^A(3)$  and  $u_j^B = Q^B(3)$ .

$row_{2j-1}$	$u_j^A$	$u_j^B$
2	$Q^A(2)$	$Q^B(2)$
3	$Q^A(3)$	$Q^B(3)$
...	...	...
2n	$Q^A(2n)$	$Q^B(2n)$
2n+1	$Q^A(2n+1)$	$Q^B(2n+1)$

$$\forall j \in \{1, \dots, n\} \quad (3)$$

Constraints (4) and (5) are used to compute the volume of *A-grapes* harvested from the beginning of the harvest.

$$Ac_0 = 0 \quad (4)$$

$$Ac_j = Ac_{j-1} + tub_j \times U_j^A \quad \forall j \in \{1, \dots, n\} \quad (5)$$

The Boolean constraint (6) indicates if the contract of *A-grapes* is satisfied.

$$contract_j = (Ac_j \geq Rmin) \quad \forall j \in \{1, \dots, n\} \quad (6)$$

Constraints (7), (8) and (9) compute the quantity of grapes in each hopper since the last emptying in the bin.

$$U_1^a = u_1^A, U_1^b = u_1^B \quad (7)$$

$$U_j^a = (1 - tub_{j-1}) \times U_{j-1}^a + u_j^A \quad \forall j \in \{2, \dots, n\} \quad (8)$$

$$U_j^b = (1 - tub_{j-1}) \times U_{j-1}^b + u_j^B \quad \forall j \in \{2, \dots, n\} \quad (9)$$

Constraint (10) represents the maximal capacity of two hoppers. Constraint (11) deals with the possibilities to fill hoppers. It depends on threshold.

$$U_j^a + U_j^b \leq 2 \times Cmax \quad \forall j \in \{1, \dots, n\} \quad (10)$$

$$(contract_j = 0) \text{ IMPLIES } U_j^a \leq Cmax \text{ AND } U_j^b \leq Cmax \quad \forall j \in \{1, \dots, n\} \quad (11)$$

We consider the distance between two extremities  $p, q \in \{2, \dots, 2n+1\}$  of two different rows (or the bin). We suppose that the machine starts and finishes harvest from the bin.

$row_{2j}$	$tub_j$	$row_{2j+1}$	$\Delta_j$
p	0	q	$d(p, q)$
p	1	q	$d(p, 1) + d(1, q) + \delta$

$$\forall j \in \{1, \dots, n\} \quad (12)$$

*sum* is a global constraint that computes the sum of the values of the set of variables. We add a new variable  $K$  that represent the total time of harvesting in the field  $D(K) = \mathbb{N}$ . We minimize  $K$  for minimizing the total time of harvesting.

$$sum(\Delta_1, \dots, \Delta_{n-1}) < K \quad (13)$$

### Related work

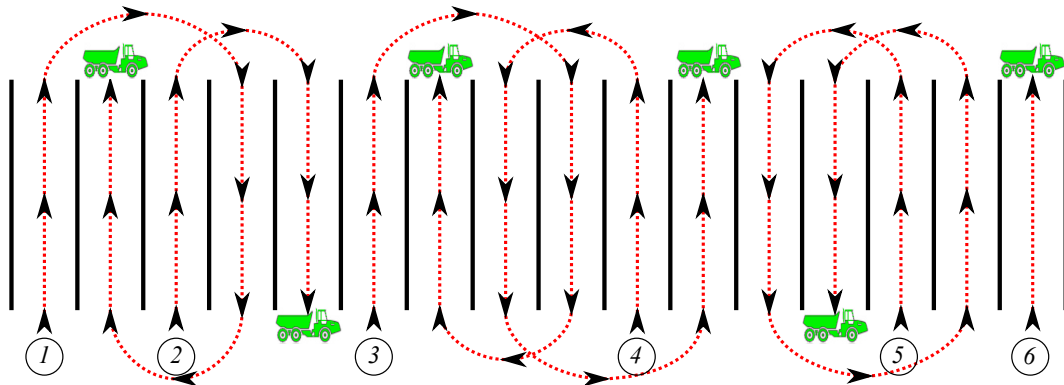
Differential Harvester Problem is close to another problem named *Vehicles Routing Problem (VRP)*. First proposed by Dantzing and Ramsey in 1959, the vehicle routing problem consists of seeking a number of customers with a fleet of vehicles. Several variants of VRP exist like VRP with time-window, VRP with different capacity on vehicles or VRP with pickup and delivery. Generally, VRPs are solved with integer linear programming formulations. This method requires a specific model for each




variant of problem in terms of linear constraints. In Bochtis & Sørensen (2009), the authors give a VRP formulation of many problems in field logistics but none of them is close enough to the Differential Harvester Problem. Thus, a constraint programming approach is interesting on such emerging problems. Changing the model only consists in adding or removing some constraints.

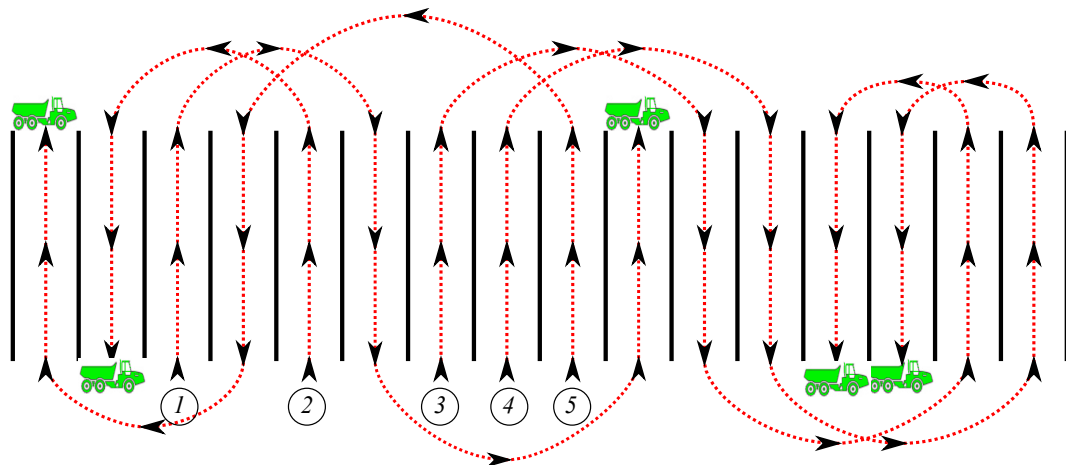
## Results


The time associated with a common harvesting approach that is illustrated by Figure . This route requires 6 emptying and takes 2307 s.



 to the bin  
Figure 4. Common harvesting route.

A solution of the problem presented in this paper is a sequence of extremities that the harvesting machine must follow. For each extremity, the solution specifies if the machine has gone to the bin. The model has been implemented in *Abscon*, a generic Constraint Programming solver that has been developed in Java by Lecoutre & Tabary (2007). With a basic implementation, we compute the optimal solution in 6 days. The optimal route is presented in Figure . It requires 5 emptying and takes 1378 s.



 to the bin  
Figure 5. Optimal route solution.

## Discussion

With a basic implementation of our model in the generic solver *Abscon*, we realized an

experimental test on real data. The solution obtained by the CP model is 40% better than the common path in terms of travel time. Also, the number of bin emptyings is lower. Of course, these results must be confirmed on more vine fields and with a larger number of rows in fields. The time to solve the problem is currently not optimized because the goal of this paper was to present our approach to the differential harvesting problem. In future work, we plan to improve the Constraint Programming model to improve computation time.

As discussed above, a Constraint Programming approach generally helps to deal with changes in the model. Adding or removing some constraints can be done without compromising the model. Hence, one might consider changing some parameters of the problem. For example, it is possible to maximize the quantity or optimize the time to harvest *A-grapes* before a given threshold or impose differential harvest with three hoppers or more. In our study, we considered that the bin is located near the plot. This is not always the case. Sometimes the bin can move along one side of the field. This possibility makes the problem more complex. Other machines, which rely on bins displaced by tractors, can sort more than two types of grape qualities. In such a case, there is no longer any capacity constraint and the problem becomes a simple shortest path search. Finally, it was assumed that the quality zones as well as the yield were known precisely. This is of course not the case, future developments will be necessary to adapt the approach to the uncertainty associated with within-field yield estimation as well as uncertainty associated with within-field quality zones.

## Conclusion

In this paper, we proposed a formal description for modeling the Differential Harvesting Problem. We used the paradigm of Constraint Programming to solve this problem. We have implemented the model with the generic solver Abscon. Preliminary experimental results on real data from a French vineyard show a 40% decrease of the harvesting time. It encourages future research to improve the Constraint Programming model in order to reduce the computation time.

## References

- Bessiere, C. 2006. Constraint Propagation. Chap. 3 of the Handbook of Constraint Programming, F. Rossi, P. van Beek and T. Walsh (Eds). Elsevier Science Inc. New York, NY, USA, pp. 29-83.
- Bochtis, D.D., Sørensen C.G. 2009. The vehicle routing problem in field logistics part I. Biosystems engineering **104** 447-45.
- Bochtis, D.D., Sørensen C.G. 2010. The vehicle routing problem in field logistics part II. Biosystems engineering **105** 180-188.
- Bramley, R. G. V.; Hamilton, R. P., 2004. Understanding variability in winegrape production systems 1. Within vineyard variation in yield over several vintages. Australian Journal of Grape and Wine Research **1** 32-45.
- Dantzig, G.B., Ramser, J.H. 1959. The truck dispatching problem. Management science **6**(1) 80-91.
- Lamb, D.W., Weedon, M.M y Bramley, R.G.V. 2004. Using remote sensing to predict grape phenolics and colour at harvest in a Cabernet Sauvignon vineyard: Timing observations against vine phenology and optimising image resolution. Australian Journal of Grape and Wine Research **10** 46–54.

- Merchez, S, Lecoutre, C., Boussemart, F., 2001. AbscCon: a prototype to solve CSPs with abstraction. 7<sup>th</sup> International Conference on Principles and Practice of Constraint Programming (CP'01). LNCS 2239, Springer 730-744.
- Tisseyre, B., Ojeda, H. y Taylor, J. 2007. New technologies and methodologies for site-specific viticulture. *Journal International des Sciences de la Vigne et du Vin* **41** 63-76.
- Urretavizcaya, I., Santesteban, L. G., Tisseyre, B., Guillaume, S., Miranda, C., & Royo, J. B. (2013). Oenological significance of vineyard management zones delineated using early grape sampling. *Precision Agriculture* **15**(1), 111–129.