

# Nouvelle borne pour le PROBLÈME DE VENDANGE SÉLECTIVE

G. Volte<sup>1</sup>, É. Bourreau<sup>2</sup>, R. Giroudeau<sup>2</sup>, O. Naud<sup>3</sup>

<sup>1</sup> LAAS, CNRS, Toulouse, France [gvolte@laas.fr](mailto:gvolte@laas.fr)

<sup>2</sup> LIRMM, Univ Montpellier, CNRS, Montpellier, France [{giroudeau,bourreau}@lirmm.fr](mailto:{giroudeau,bourreau}@lirmm.fr)

<sup>3</sup> ITAP, Univ Montpellier, INRAE, Institut Agro, Montpellier, France [olivier.naud@inrae.fr](mailto:olivier.naud@inrae.fr)

## 1 Définition du problème

Le DIFFERENTIAL HARVEST PROBLEM consiste à optimiser le temps de récolte d'une parcelle viticole en triant dans une benne spécifique une quantité minimum donnée de raisins de meilleure qualité (voir [3] pour plus de détails).

Grâce à une information agronomique obtenue a priori, il est possible de cartographier (cf. Figure 1) les rangs de la parcelle en distinguant des zones en fonction de la qualité des raisins à vendanger. Ce problème a été introduit par [1], qui a employé la *programmation par contraintes* pour le résoudre, puis a été étudié par [2], via une approche de *model checking*.



FIG. 1 – Figure illustrant les différentes zones de qualité de vigne.

L'objectif du problème est de minimiser le temps de récolte du champ tout en récoltant une certaine quantité ( $Rmin$ ) de raisins de bonne qualité.

La direction de récolte des rangs est contraignante pour satisfaire la contrainte  $Rmin$ . Lorsque la valeur de  $Rmin$  est élevée il faut s'assurer qu'un certain nombre de rangs soit récolté dans le *bon sens* (celui qui maximise la récolte de raisin  $\mathcal{A}$ ).

Nous présentons une borne sur un nombre de rangs particuliers (les *rangs limitants*) : les rangs devant, tous, être récoltés dans le même sens (leur *bon sens*) sans pouvoir les associer avec des rangs récoltés dans l'autre direction. Cette borne ajoutera, dans nos modèles, l'information sur la direction d'un sous-ensemble de rangs lorsque la valeur de  $Rmin$  est élevée.

## 2 Propriétés sur la direction de récolte des rangs

Nous définissons  $N^{gd} = \{i \in N, q_i^{gd}(\mathcal{A}) > q_i^{dg}(\mathcal{A})\}$  (resp.  $N^{dg} = \{i \in N, q_i^{gd}(\mathcal{A}) < q_i^{dg}(\mathcal{A})\}$ ) comme l'ensemble des rangs devant être récolté de gauche à droite (resp. droite à gauche) pour maximiser la récolte de raisin  $\mathcal{A}$  et  $N^= = \{i \in N, q_i^{gd}(\mathcal{A}) = q_i^{dg}(\mathcal{A})\}$  comme l'ensemble des rangs pour lesquels la récolte est symétrique avec  $n = |N^{gd}| + |N^{dg}| + |N^=| = n^{gd} + n^{dg} + n^=$ . Le nombre de rangs asymétriques pouvant être récoltés dans les deux sens sans violer la contrainte  $Rmin$  est donné par  $w$ . La notion de rangs limitants est fondamentale pour le PVS, elle permet notamment de mesurer l'influence des directions de récolte des rangs asymétriques sur la contrainte  $Rmin$ .

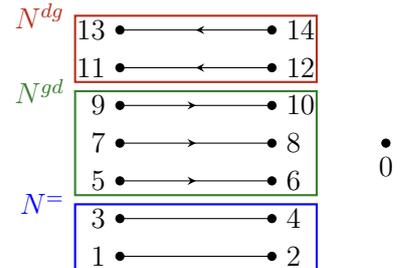


FIG. 2 – Illustration des ensemble de rangs.

L'idée générale est de former le plus d'aller-retour sans récolter de rangs dans le *mauvais sens* en se servant des rangs symétriques et de la complémentarité entre les rangs asymétriques, puis d'utiliser les  $w$  déviations.

**Théorème 1**

Le nombre de rangs limitants, noté  $n_{lim}$ , présents dans toute solution réalisable est de  $n_{lim} = |n^{gd} - n^{dg}| - n = -2 * w$ .

Tout rang limitant doit être récolté par un arc diagonal (un arc représentant le trajet de la machine traversant le champ sans récolte de raisins). Pour renforcer la relaxation linéaire de nos modèles nous ajoutons la contrainte suivante aux modèles avec  $\mathcal{D}$  représentant l'ensemble des arcs diagonaux .

$$\sum_{a \in \mathcal{D}} x_a^s \geq n_{lim} \tag{2.1}$$

### 3 Influence de la borne inférieure du nombre d'arcs diagonaux

Nous montrons empiriquement les bénéfices de cette borne sur la résolution des instances avec une valeur de *Rmin* élevée pour des modèles utilisant la PL développés pour et résolus par VRPSOLVER. Pour cette expérience, nous introduisons six classes d'instances avec *Cmax* = 1000. Considérons  $Rmin = \{95\%, 96\%, 97\%, 98\%, 99\%, 100\%\}$ . Les six classes possèdent les mêmes 65 instances aléatoirement sélectionnées. Toutes les instances sont résolues avec et sans la contrainte (2.1) avec une limite du temps de résolution fixée à 600 secondes.

Le Tableau 1 décrit synthétiquement les expériences numériques réalisées en comparant le gap d'intégralité pour le nœud racine et le temps de résolution (solution optimale entière) moyen pour toutes les instances de chaque classe et en calculant le ratio entre les valeurs obtenues avec et sans la contrainte (2.1).

| <i>Rmin</i> (%) | sans (2.1) |        | avec (2.1) |        | gap    | tps    |
|-----------------|------------|--------|------------|--------|--------|--------|
|                 | gap(%)     | tps(s) | gap(%)     | tps(s) |        |        |
| 95              | 2,56       | 67,07  | 1,35       | 12,74  | *1,89  | *5,26  |
| 96              | 8,78       | 201,58 | 1,15       | 6,50   | *7,65  | *31,00 |
| 97              | 14,43      | 307,65 | 0,73       | 5,12   | *19,77 | *60,03 |
| 98              | 13,26      | 321,83 | 0,59       | 4,86   | *22,52 | *66,26 |
| 99              | 10,20      | 374,97 | 0,25       | 4,76   | *41,16 | *78,79 |
| 100             | 0,29       | 3,81   | 0,12       | 2,44   | *2,15  | *1,56  |

TAB. 1 – Synthèse des résultats sur l'influence de la contrainte (2.1) sur la relaxation linéaire.

En moyenne, sur toutes les instances testées, le facteur de gain sur le gap d'intégralité est de \*18,06 et sur le temps de \*25,80, ce qui représente une augmentation considérable des performances du modèle avec cette unique contrainte supplémentaire.

## Références

[1] N. BRIOT, C. BESSIERE et P. VISMARA. « A Constraint-Based Approach to the Differential Harvest Problem ». In : *Principles and Practice of Constraint Programming*. Sous la dir. de Gilles PESANT. Cham : Springer International Publishing, 2015, p. 541–556. ISBN : 978-3-319-23219-5.

[2] R. SADDEM-YAGOUBI et al. « Model-checking precision agriculture logistics : the case of the differential harvest ». In : *Discrete Event Dynamic Systems* 30 (2020), p. 579–604.

[3] Gabriel VOLTE et al. « Using VRPSolver to efficiently solve the Differential Harvest Problem ». In : *Comput. Oper. Res.* 149 (2023), p. 106029. DOI : 10.1016/j.cor.2022.106029. URL : <https://doi.org/10.1016/j.cor.2022.106029>.