# *VRPSolver* pour résoudre le problème de vendange sélective

G. Volte[1], É. Bourreau[1], R. Giroudeau[1], O. Naud[2]

[1] LIRMM, Univ Montpellier, CNRS, Montpellier, France {volte,giroudeau,bourreau}@lirmm.fr

[2] ITAP, Univ Montpellier, INRAE, Institut Agro, Montpellier, France olivier.naud@inrae.fr

## 1 Définition du problème

Le Problème de Vendange Sélective 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 [4] pour plus de détails).

Grâce à une information agronomique obtenue à priori, il est possible de cartographier (cf. Figure 1) les $|N|$ 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 [3], via une approche de *model checking*.

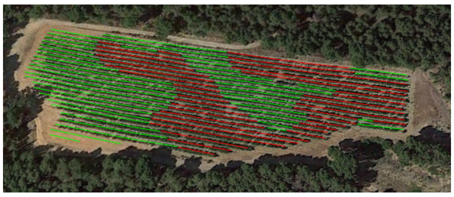

FIG. 1 – Figure illustrant les différentes zones de qualité de vigne.

Nous présenterons notre approche basée sur l'utilisation d'un solveur spécifique pour le VRP : *VRPSolver* [2], ainsi que des résultats comparatifs avec d'autres méthodes de résolution.

## 2 Un solveur spécifique : *VRPSolver*

VRPSolver [2] est un solveur générique de problèmes de tournées de véhicules et traite également des problèmes de bin-packing, vector packing et Generalized Assignment Problem (GAP), offrant un éventail de techniques d'optimisation présentes dans la littérature, développées spécifiquement pour certaines variantes de problèmes et permettant d'obtenir des solutions équivalentes aux meilleures méthodes de résolution présentes dans l'état de l'art. Nous présentons ici, les subtilités de modélisation pour utiliser *VRPSolver* afin de résoudre efficacement le problème de vendanges sélective.

### 2.1 Modélisation du Problème de Vendange Sélective avec *VRPSolver*

On peut commencer par remarquer que le Problème de Vendange Sélective se comporte comme un problème de tournées de véhicules hétérogènes. En effet, la vendangeuse fonctionne suivant 2 modes : sélectif $s$ (tri des raisins de bonne qualité dans l'une des trémies) et non-sélectif $\overline{s}$. Définissons $K = \{s, \overline{s}\}$, comme l'ensemble des types de véhicules.

Chaque rang est décomposé en deux sommets, ses extrémités, et le choix de la première extrémité visitée indique le sens de parcours du rang. Nous utilisons un ensemble de deux graphes $G^k = (V^k, A^k)$ pour modéliser l'espace de recherche des routes de type $k$. Dans chaque graphe, la source et le puits sont représentés par le même sommet $v^k_{source} = v^k_{sink} = 0$, $\forall k \in K$, ce sommet correspond au dépôt.

La ressource $A_s$ (resp. $B_s$) calcule la récolte de raisins $\mathcal{A}$ (resp. $\mathcal{B}$) pour une route sélective. Pour les routes non sélectives, les deux qualités de raisins sont mélangées en une seule, l'ensemble des ressources est $R_{\overline{s}} = \{B_{\overline{s}}\}$. La consommation des ressources sur chaque arc est donnée par $q(a, A_s) = A_i,\ q(a, B_s) = B_i$ pour les routes sélectives, et $q(a, B_{\overline{s}}) = A_i + B_i,\ a = (i, j) \in A^{\overline{s}}$ pour les routes non sélectives.

Pour les routes sélectives, les deux trémies sont considérées séparément et chacune à une capacité de $Cmax$, donc l'intervalle de consommation $[l_{i,A_s}, u_{i,A_s}]$ (resp. $[l_{i,B_s}, u_{i,B_s}]$), défini pour tout sommet $i \in V^s$, pour la ressource $A_s$ (resp. $B_s$) est $[0, Cmax]$. Pour les routes non sélectives, les deux trémies sont fusionnées en une trémie de capacité $2 * Cmax$, ainsi pour la ressource $B_{\overline{s}}$, l'intervalle de consommation est $[l_{i,B_{\overline{s}}}, u_{i,B_{\overline{s}}}] = [0, 2 * Cmax]\ \forall i \in V^{\overline{s}}$.

Les variables binaires $x_a^k,\ \forall a \in A^k, \forall k \in k$ indiquent si l'arc $a$ est utilisé dans le graphe $G^k$. Le nombre minimum et maximum de véhicules sélectifs est donné par $L^s = \left\lceil \frac{Rmin}{Cmax} \right\rceil,\ U^s = |N|$. Le nombre minimum et maximum de véhicules non-sélectifs est donné par $L^{\overline{s}} = 0,\ U^{\overline{s}} = |N|$. Le mapping entre les variables de décision et les arcs est donné par la fonction $M : M(x_a^k) = a,\ \forall a \in A^k, k \in K$.

Le packing set est l'outil générique introduit par les auteurs de *VRPSolver* [2] afin de modéliser les objets atomiques (clients, rangs, etc.) d'un problème. Plusieurs sommets d'un même packing set ne peuvent appartenir à une solution. Le packing set pour le Problème de Vendange Sélective est défini par $\mathcal{B}^{\mathcal{V}} = \cup_{i \in N} \{\{v_{2*i}^k, v_{2*i+1}^k : k \in K\}\}$. La contrainte disjonctive entre les extrémités d'un rang est ainsi modélisée.

La fonction objectif (1) est de minimiser le coût total des routes. La fonction $\delta^-(i, j)$ donne l'ensemble des arcs entrants dans $i$ ou $j$, ainsi la contrainte (2) vérifie que chaque rang est traversé une seule fois. Le contrainte (3) assure que le seuil *Rmin* de raisins $\mathcal{A}$ est récolté.

Modèle 1: Problème de Vendange Sélective avec VRPSolver

$$Min \quad \sum_{k \in K} \sum_{a \in A^k} c_a^k x_a^k \tag{1}$$

$$S.t \quad \sum_{k \in K} \sum_{a \in \delta^-(i,j)} x_a^k = 1 \qquad \forall (i, j) \in N \tag{2}$$

$$\sum_{(i,j) \in A^s} A_i * x_{(i,j)}^s \geq Rmin \tag{3}$$

$$x_a^k \in \{0, 1\} \qquad \forall k \in K, a \in A^k \tag{4}$$

## Références


[1] N. Briot, C. Bessiere, and P. Vismara. A constraint-based approach to the differential harvest problem. In Gilles Pesant, editor, *Principles and Practice of Constraint Programming*, pages 541–556, Cham, 2015. Springer International Publishing.

[2] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. A generic exact solver for vehicle routing and related problems. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 354–369. Springer, 2019.

[3] Rim Saddem-Yagoubi, Olivier Naud, Karen Godary-Dejean, and Didier Crestani. Model-checking precision agriculture logistics : the case of the differential harvest. *Discrete Event Dynamic Systems*, 30 :579–604, 2020.

[4] Gabriel Volte, Eric Bourreau, Rodolphe Giroudeau, and Olivier Naud. Exact method approaches for the differential harvest problem. In *17th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, volume 12296 of *Lecture Notes in Computer Science*, pages 492–510, Vienna, Austria, September 2020.