



## Strategic Planning of Phytosanitary treatments in Wineries

Agnès Le Roux, Boris Detienne, Ruslan Sadykov, Issam Tahiri, Alexis Toullat,  
François Vanderbeck

### ► To cite this version:

Agnès Le Roux, Boris Detienne, Ruslan Sadykov, Issam Tahiri, Alexis Toullat, et al.. Strategic Planning of Phytosanitary treatments in Wineries. ROADEF, Feb 2016, Compiègne, France. hal-01425704

**HAL Id: hal-01425704**

**<https://inria.hal.science/hal-01425704>**

Submitted on 3 Jan 2017

**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.

# Strategic Planning of Phytosanitary treatments in Wineries

Agnès Le Roux<sup>1,2</sup>, Boris Detienne<sup>1,2</sup>, Ruslan Sadykov<sup>2,1</sup>,  
Issam Tahiri<sup>1,2</sup>, Alexis Toullat<sup>1</sup>, François Vanderbeck<sup>1,2</sup>

<sup>1</sup> Université de Bordeaux, Institut de Mathématiques de Bordeaux  
, 351 cours de la Libération, 33405 Talence, France

{agnes.le-roux,boris.detienne,issam.tahiri,alexis.toullat,fv}@math.u-bordeaux1.fr

<sup>2</sup> INRIA Bordeaux – Sud-Ouest

200 avenue de la Vieille Tour, 33405 Talence, France {ruslan.sadykov}@inria.fr

**Mots-clés :** *Column generation, extended formulation, Dynamic programming, phytosanitary treatments*

## 1 Introduction and Problem Definition

We consider planning phytosanitary treatments in a vineyard. We are given a set of diseases (or requests)  $r \in R$  that must be treated for each site  $s \in S$ . Product mixtures  $p \in P$  are defined by their composition of active components  $c \in C$ , and their duration,  $D_{pr}$ , of protective power for each request  $r$ . Machines  $m \in M$  are available to spread the mixtures on the sites. The time horizon is divided in time periods  $t \in T$ . Sites are partitioned in sectors  $\sigma \in \Sigma$ . The objective of the problem is to minimize the machine leasing costs, their travel cost to sectors and the costs related to the product use. The following constraints must be enforced. On each site :

- every request must be covered by a treatment at any time ;
- no more than one treatment can be applied in any time period ;
- the cumulative amount of toxic components applied is bounded ;
- the number of application of a component is restricted for some toxic components ;
- if one product mixture is spread, the following mixture must be a possible successor of the one used before.

Over the whole vineyard :

- the number of journeys to a sector is determined by the machine tank capacity ;
- the total occupation of a machine (counting the time needed for traveling and for spreading tasks) is restricted for each period.

## 2 Column Generation, pricing using Dynamic Programming

To solve this problem, we use a column generation approach where the machine policy and the product order policy are pure master decisions, while treatment planning decisions are made in individual pricing subproblems associated with each site. We developed a dedicated dynamic program to solve the pricing subproblems.

Our first pricing oracle variant is a forward labeling algorithm. In this dynamic programming approach, a partial solution is associated to a sub-path in the state graph, where nodes represent states of the system, and are defined by a label  $L$ . A partial solution can be extended by deciding what treatment to do next with what machine. Such transitions are associated to arcs in the state graph. To identify feasible transitions (possible label extensions for  $L$ ), we need to compute the set of diseases  $R^L$  that must be treated (a request must be treated if the last treatment for it does not cover it anymore). Then, we compute the set of mixtures

$P^L$  that can be applied (they must treat all the diseases of  $R^L$ , must respect the consecutive treatments constraints and must not induce an overuse of restricted components). Finally, we compute the set of machines  $M_p^L$  that can be used to spread the mixture  $p \in P^L$  (the machine must not consume too much mixtures to avoid exceeding the limit of toxic components spread on the site). The dominance rule must be carefully defined to limit the number of intermediate partial solutions that are kept alive while guaranteeing to maintain a path to the optimum. (In practice, about 80% of the labels are cut by dominance.)

Our second variant is a backward recursion. Compared to the forward labeling procedure, it tends to generate fewer feasible partial solutions because the resource constraints (toxicity and restricted components) are tighter towards the end of the time horizon. The backward approach requires a careful definition of feasible transitions, as the protection duration of a mixture depends on the date at which it is applied, transitions that do not cover pending diseases are admitted provided there exists other covering options in earlier time periods.

We then consider further restrictions on the decision space to speed-up pricing. The first restriction is to forbid treatments that are too many periods ahead of the date by which the next treatment must be applied (the idea is to avoid over-coverage of a request). Two parameters  $\alpha$  and  $\beta$  are set; they respectively represent the maximum number of time periods ahead to trigger a preventive treatment and the total budget of anticipation. We propose a second restriction where the set of diseases is partitioned into independent request families such that each time a mixture is applied, it treats all (and only) the requests of one family. This latter restriction gives rise to independent flow models for each request family.

### 3 Experimental Tests and Conclusion

We performed tests on a real data set with about 170 sites, 5 requests, 8 machines, 16 active components and 17 product mixtures over a horizon of 73 periods. We compare a direct approach (with a compact formulation) with a column generation approach where the pricing problem is either solved with a commercial MIP solver (Cplex 12.6), the forward labeling algorithm or the backward labeling algorithm. We solve continuous relaxation of the extended model by column generation and then use mip heuristics based on column generation to obtain a primal solution. We obtained solutions for the full instance using column generation and primal heuristics that are of significantly better than those obtained with a direct mip approach (better objective values and shorter computing times).