Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

A robust optimization approach to wine grape harvesting scheduling

Carlos Bohle, Sergio Maturana *, Jorge Vera

Industrial and Systems Engineering Department, Escuela de Ingenieria, Pontificia Universidad Católica de Chile, Casilla 306, Correo 22, Santiago, Chile

ARTICLE INFO

ABSTRACT

Optimization models are increasingly being used in agricultural planning. However, the inherent uncertainties present in agriculture make it difficult. In recent years, robust optimization has emerged as a methodology that allows dealing with uncertainty in optimization models, even when probabilistic knowledge of the phenomenon is incomplete. In this paper, we consider a wine grape harvesting scheduling optimization problem subject to several uncertainties, such as the actual productivity that can be achieved when harvesting. We study how effective robust optimization is solving this problem in practice. We develop alternative robust models and show results for some test problems obtained from actual wine industry problems.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Optimization models have been used for many years to support decision making in different fields. Applications in other industries abound in the literature, particularly in the area of production planning and scheduling [2,11,15,17]. In agriculture the application of optimization models has been less successful. This is probably due to cultural aspects in this area as well as its inherent complexity. However, there has been a steady development in the last several years, as can be seen in [1].

This work addresses an important part of the complexity, namely the uncertainty, in the rapidly growing Chilean wine industry. In a previous research, Ferrer et al. [14] present an optimization model to help schedule harvesting operations in a vineyard and assigning labor to operations. This problem is important since harvesting the grapes at the right time is crucial for the quality of the wine. The model described in [14] takes into account, not only the direct costs involved in the process, but also the potential cost of quality deterioration due to harvesting before or after the optimal ripeness date. This model balances the operational costs and quality costs, depending on the expected quality of the final wine. However, there are important sources of uncertainty, which are inherent to the process. In this paper, we propose a methodology to handle some of those uncertainties, even when only a limited knowledge of their characteristics is available.

In general, optimization models in most applications use data which is subject to uncertainty and measurement errors. The effect of this uncertainty must be addressed. This is particularly important in some supply chain management problems where the cost of making a wrong decision can be high. The wine grape harvesting

problem we address in this paper is an example of this, since bad decisions may result in downgrading a premium quality wine to a lower quality wine, which will have to be sold at a much lower price.

There are different approaches for dealing with uncertainty. One of them is to simply ignore it and solve the problem using "average" values, or "most probable" values. Another approach is to explicitly model uncertainty into the problem and to consider these elements in the constraints and objective function, together with some form of distributional information. This is addressed by the whole field of stochastic programming (see [10,19], for example). However, in many applications, full distributional information is unknown or very hard to determine.

A more recent approach, robust optimization, attempts to compute feasible solutions for a whole range of scenarios of the uncertain parameters, while optimizing an objective function in a controlled and balanced way with respect to the uncertainty in the parameters. The concept of robust optimization has been used by Mulvey et al. [22] when addressing stochastic problems in which solutions must be feasible for a discrete set of probabilistic scenarios. An application of this approach in the logistic operations of a wine company is described by Yu and Li [26]. The approach we use in this work has been developed by Ben-Tal and Nemirovski [4] following related developments from the robust control area. In this case, data is assumed to belong to some set with no specific probabilistic information.

There have been several studies on how to deal with different types of uncertainties in agriculture. For example, Allen and Schuster [3] developed a mathematical model to control harvest risk, in the case of the Concord grapes. The focus of this model is on balancing the risk of overinvestment with the risk of underproduction. Kazaz [18] studied production planning with random yield and demand in the case of olive oil production. Two uncertain

* Corresponding author. Tel.: +56 2 354 4272; fax: +56 2 552 1608.
  E-mail address: smaturan@ing.puc.cl (S. Maturana).

elements were considered: olive production yield and olive oil demand.

The structure of the paper is as follows. In Section 2, we discuss robust optimization and the various modeling approaches presented in the literature. In Section 3, we present the specific robust model we will apply to the grapes harvesting problem. In Section 4, we present some computational results which show how to analyze the performance of the robust methodology and how to interpret the results. We finally close with our conclusions in section 5. The harvesting scheduling model presented in [14] is summarized in the Appendix. Although the model used in this paper was developed for the wine industry, we believe that this situation is typical of many agricultural areas, such as fruit collection and livestock management.

2. Robust optimization

The robust optimization methodology used in this paper is a way of dealing with variability in the data of an optimization problem. A solution to a problem instance is called "robust" if it is insensible to perturbations of the data, at least within a certain range. A robust solution is feasible for different scenarios of the data, and as such is not necessarily optimal for any one of them. Consider the linear programming problem:

$$P) \quad \begin{aligned} z^* = \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leqslant b, \quad x \geqslant 0, \end{aligned} \tag{1}$$

and furthermore suppose that the data for this problem, $d = (A, b, c)$, is not known exactly. The only fact we really know is that $d \in S$, where S is a certain (closed and bounded) set which reflects our knowledge of the problem data. A robust optimization formulation corresponds to the following problem:

$$PR) \quad \begin{aligned} z_R = \min \quad & t \\ \text{s.t.} \quad & c^T x \leqslant t, \\ & Ax \leqslant b, \quad x \geqslant 0, \\ & \forall (A, b, c) \in S. \end{aligned} \tag{2}$$

Let x^* be an optimal solution to PR). x^* is a robust solution in the sense that is feasible for *all* occurrences of the data. The loss due to the use of a robust solution is measured by the objective function deterioration $|z^* - z_R|$. However, z^* is not known, as it corresponds to the optimal value of whatever will be the actual occurrence of the data. This robust solution is, in a sense, a worst case solution and it could be very pessimistic. Soyster [24] proposed an approach to handle column uncertainty in intervals, in which the worst possible outcome for each data component is assumed. In fact, "box uncertainty" is one of the simplest representations. For this we mean that

$$(A, b, c) \in S \Longleftrightarrow \underline{A} \leqslant A \leqslant \overline{A}, \quad \underline{b} \leqslant b \leqslant \overline{b}, \quad \underline{c} \leqslant c \leqslant \overline{c}, \tag{3}$$

where the inequalities are to be interpreted componentwise. In the second half of the 1990s, Ben-Tal and Nemirovski [4–6] and, independently, El Ghaoui et al. [12,13] proposed using different formulations of PR). They show that the robust reformulation of an optimization problem with uncertainty is itself an optimization problem whose structure depends on the specification of the uncertainty set S. This is important as problem PR) stated above seems to be a semi-infinite optimization problem, which is, in general, a very hard problem to solve. In the case of linear programming with box uncertainty, as described above, the robust equivalent problem is itself a linear program.

Since box uncertainty leads to a worst case scenario, Ben-Tal and Nemirovski have considered a form of uncertainty which restricts the data to an ellipse. Ellipsoidal uncertainty tends to better represent the possible interactions among the different data parameters and potentially avoids the worst case scenario (which is very unlikely). Ellipsoidal uncertainty is described as follows, for the case of variation in the matrix A coefficients: if α_i is the i-th row of the matrix A, we define the "uncertain" set S_i as:

$$S_i = \{\alpha : (\alpha - \alpha_i^0)^T V_i^{-1} (\alpha - \alpha_i^0) \leqslant r^2\}, \tag{4}$$

where α_i^0 is a central or "nominal" value and V_i is a positive definite matrix, which can be thought of as a kind of "covariance" matrix. Recall, however, that no explicit probabilistic information is being used here. This model allows representing more complex interactions between the data elements than the box uncertainty model. Using a model of ellipsoidal uncertainty, the robust equivalent becomes a conic quadratic optimization problem. The fact that many robust reformulations correspond to a convex conic (or explicitly semidefinite) problems is important as this leads to the solution of robust optimization problems using interior point methods for convex optimization, as studied by Nesterov and Nemirovski [23].

More recently Bertsimas and Sim [7,8] have provided an alternative way of representing parameter uncertainty, which leads to a different robust counterpart. This proceeds by bounding the number of parameters in the data subject to variability, and the amount of variability. For instance, suppose that only the matrix A is allowed to vary, and suppose that a_{ij} denotes a coefficient of the matrix. Suppose that a_{ij} varies with respect to a central (nominal) value a_{ij}^0 in an amount of at most s_{ij}. Then the uncertainty set is given by

$$S = \left\{ A \in R^{m \times n} : a_{ij} \in [a_{ij}^0 - s_{ij}, a_{ij}^0 + s_{ij}], \forall i, j; \sum_{j=1}^{n} \frac{|a_{ij} - a_{ij}^0|}{s_{ij}} \leqslant \Gamma, \forall i \right\}. \tag{5}$$

The parameter Γ is considered by Bertsimas and Sim as an "uncertainty budget". If it is large (close to n or the number of uncertain parameters in a constraint), the entire variation in all parameters might be achieved, being equivalent to the worst case scenario described by Soyster. If it is small, simultaneous large variations in many parameters are excluded. This is justified under the argument that worst case scenarios are very unlikely. Note also that the whole normalized variation is being measured in the L_1 norm. In fact, Bertsimas and Sim have identified the measure of the uncertainty set as a combined norm, which is denoted as the $L_1 \cap L_\infty$ norm.

Since using absolute values leads to linear equivalents, this formulation of robustness has the advantage of producing an equivalent robust problem of the same kind as the original one. That is, if the constraints are linear, this will produce a linear robust transformation. In particular, this allows handling mixed integer problems, as the integrality constraints are preserved without being affected by the robust transformation. Although this is also true in the previously mentioned general approach of Ben-Tal and Nemirovski, under ellipsoidal uncertainty this will lead to a nonlinear mixed integer problem, which in general is hard to solve. In the Bertsimas and Sim approach the robust equivalent is still a mixed integer linear problem. Specifically, for a linear programming problem of the form

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leqslant b \\ & x \geqslant 0 \end{aligned} \tag{6}$$

the robust counterpart is constructed by considering a protection function for each constraint, which results in the following problem:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & (\alpha_i^0)^T x + \beta_i(x, \Gamma_i) \leqslant b_i \quad i = 1, \ldots, m, \\ & x \geqslant 0, \end{aligned} \tag{7}$$

where α_i^0 denotes the nominal data for row i and where the protection function for each constraint $i = 1, \ldots, m$ is:

$$\beta_i(x, \Gamma_i) = \max \quad \sum_{j=1}^n x_j s_{ij} u_{ij}$$
$$\text{s.t.} \quad \sum_{j=1}^n u_{ij} \leqslant \Gamma_i, \tag{8}$$
$$0 \leqslant u_{ij} \leqslant 1 \quad j = 1, \ldots, n.$$

Note that it is possible to use different uncertainty budgets for each constraint. The corresponding robust reformulation is obtained by injecting the dual of (8) onto (7) leading to:

$$\max \quad c^T x$$
$$\text{s.t.} \quad \sum_{j=1}^n a_{ij}^0 x_j + z_i \Gamma_i + \sum_{j=1}^n p_{ij} \leqslant b_i \quad i = 1, \ldots, m,$$
$$z_i + p_{ij} \geqslant s_{ij} y_j \quad i = 1, \ldots, m; \ j = 1, \ldots, n, \tag{9}$$
$$-y_j \leqslant x_j \leqslant y_j \quad j = 1, \ldots, n,$$
$$(x_j, p_{ij}, y_j, z_i) \geqslant 0 \quad i = 1, \ldots, m; \ j = 1, \ldots, n.$$

As can be seen, this is a linear program. The additional variables z_i and p_{ij} appear here as dual variables of the corresponding constraints of the protection problem.

Recently, an iterative solution approach developed by Bienstock and Ozbay [9] has been used to find robust solutions to some supply chain management problems. It has the potential of allowing more flexibility on the characteristics of the uncertainty in the data and adapts better to different models. This approach can be interpreted as using Bender's decomposition, or a cutting planes method to the general robust counterpart (PR). In this approach, only a finite set of scenarios is considered at each iteration, which solves a problem of the form:

$$\max \quad c^T x$$
$$\text{s.t.} \quad A_k x \leqslant b \quad k = 1, \ldots, p, \tag{10}$$
$$x \geqslant 0,$$

where $A_k \in S, k = 1, \ldots, p$. After solving this problem and obtaining a solution x^*, a corresponding "adversarial" problem is solved. This problem corresponds to solving the m protection function problems (8) for the current solution x^*, or, equivalently, the following single problem, which we call the adversarial problem:

$$\max \quad \sum_{i=1}^m \beta_i(x^*, \Gamma_i)$$
$$\text{s.t.} \quad \beta_i(x^*, \Gamma_i) = \sum_{j=1}^n x^* s_{ij} u_{ij} \quad i = 1, \ldots, m, \tag{11}$$
$$\sum_{j=1}^n u_{ij} \leqslant \Gamma_i \quad i = 1, \ldots, m,$$
$$0 \leqslant u_{ij} \leqslant 1 \quad i = 1, \ldots, m; \ j = 1, \ldots, n.$$

By solving this problem, we are determining the maximal protection one could achieve on the constraints with respect to the current solution x^*. A solution, u^* of this problem allows constructing a new scenario, where the coefficients are of the form $a_{ij} = \bar{a}_{ij} + s_{ij} u_{ij}^*$, for which the solution x^* is not feasible. Hence, we define a new set of constraints $A_{p+1} x \leqslant b$ to be added to the "master" problem (10). We solve again the master problem to obtain a new solution and continue in this form, adding more and more scenarios of the data, until a suitable stopping criteria is satisfied. Thus, the procedure tries to iteratively obtain a feasible solution for all scenarios of the data. This adversarial approach has been used by Bienstock and Ozbay [9] to solve a base-stock inventory problem with uncertain demand.

3. Robust model

The main decision variables in the grapes harvest planning model described in [14] correspond to the number of workers to assign at different blocks and at different moments of time. Given the productivity of labor, a certain volume of grapes are harvested each day in which workers are assigned. The data of the model specify a day in which it is optimum to harvest, but it is possible to harvest any day within a given time window. If harvesting deviates from the optimal day, a cost is incurred, corresponding to the potential loss of quality of the final wine. In addition, operational costs are also incurred. The model tries to allocate harvesting so that the total cost is minimized and labor availability and processing capacity are not surpassed. The full details of the model are presented in the Appendix.

The model has been used to allocate workers and generate harvesting schedules in actual vineyards. There are several uncertain factors in the data of the model that might have a significant impact on the operations. One of these is the actual harvesting productivity, which is difficult to model analytically. Productivity variation is not only due to the inherent variation in labor execution but it can be also affected by the weather or other geographical factors. Actual yields also vary by block.

The effect of these uncertainties has been recognized in the research of agricultural supply chain, as discussed in [16]. This has also generated concerns on how to reduce variability in labor by making the job easier. For instance, ergonomic aspects of grape picking are addressed in [20]. However, specific statistical analysis and the form of the variation of labor productivity are hard to obtain. This makes the subject an interesting case for robust optimization, as we can assume different values of uncertainty and analyze the response of the model.

The constraints affected by uncertain harvest productivity are the equations represented in (A.10) in the Appendix. However, since we will focus on the most uncertain parameters, which are the productivities of the manual harvesting or handpicking method, P_{j2}, the relevant constraints are the following:

$$\sum_v x_{jt2v} \leqslant P_{j2} u_{jt2}, \quad \forall j \in J_2, \ \forall t, \tag{12}$$

where J_2 is the set of blocks to be handpicked. The constraints state that the total manual harvest at block j in day t, regardless of destination v, cannot exceed the capacity induced by using u_{jt2} workers. The problem with this constraint, as written, is that it does not fit the general setup for the robust paradigm presented in Section 2. There is only one uncertain parameter in each constraint and the only robustness that can be built in this situation, following the standard constructions is to take the worst possible value of the parameter in all constraints. This might be too conservative since it is very unlikely that all parameters be in their worst case at the same time. However, it still makes sense to find a robust solution that avoids the unlikely worst case scenario. To achieve this, we propose introducing a modification to the model, using a redundant constraint, and applying the standard robust transformation, while adding extra protection to the original constraints.

3.1. Aggregated modified robust reformulation

We now assume that the uncertain parameter has a nominal value of $\bar{P}_{j2}$ and a maximum variation expressed as a fraction of the nominal value, given by a parameter $\delta, 0 \leqslant \delta \leqslant 1$. For convenience, we will also express the uncertainty budget parameter Δ as a fraction, $0 \leqslant \Gamma \leqslant 1$, of the total number of uncertain parameters in the constraint, which is $\sum_{j \in J_2} F_{jt}$, where F_{jt} is a parameter that indicates whether it is feasible to harvest block j on day t. Under these

assumptions, and following the Bertsimas and Sim formulation, the uncertainty set for P_{j2} is given by

$$\left\{ P_{j2}, j \in J_2 : |\overline{P}_{j2} - P_{j2}| \leqslant \delta \overline{P}_{j2}, j \in J_2; \sum_{j \in J_2} \frac{|\overline{P}_{j2} - P_{j2}|}{\delta \overline{P}_{j2}} \leqslant \Gamma \sum_{j \in J_2} F_{jt} \right\}. \tag{13}$$

We begin the reformulation by incorporating into the problem aggregate constraints obtained by adding constraints (12) over the index j. This leads to the following additional constraints to be added to the model:

$$\sum_{v} \sum_{j \in J_2} x_{jt2v} \leqslant \sum_{j \in J_2} P_{j2} u_{jt2}, \quad \forall t. \tag{14}$$

Note that, although these constraints are redundant, we will use them to write the corresponding robust constraints associated with (14). This leads to:

$$\sum_{v} \sum_{j \in J_2} x_{jt2v} - \sum_{j \in J_2} \overline{P}_{j2} u_{jt2} + \upsilon_t \Gamma \left(\sum_{j \in J_2} F_{jt} \right) + \sum_{j \in J_2} \alpha_{jt} \leqslant 0 \quad \forall t,$$
$$\alpha_{jt} + \upsilon_t \geqslant \delta \overline{P}_{j2} \beta_{jt} \quad \forall t, \ \forall j \in J_2, \tag{15}$$
$$u_{jt2} \leqslant \beta_{jt} \quad \forall t, \ \forall j \in J_2.$$

In this formulation, α_{jt}, υ_t are the auxiliary dual variables used in the robust formulation. However, these robust constraints, even with $\Gamma = 1$, might not give a feasible solution for all possible occurrences of the uncertain parameter since the robustness is applied to an aggregate constraint. To overcome this problem, we propose a modified robust counterpart, which incorporates part of the dual information contained in the α_{jt} and υ_t variables onto the original constraints (12). We propose using the following robust version of (12):

$$\sum_{v} x_{jt2v} - \overline{P}_{j2} u_{jt2} + \upsilon_t \Gamma + \alpha \leqslant 0, \quad \forall t, \forall j \in J_2. \tag{16}$$

Note that, for each index j, this constraint can be interpreted as "desegregating" the robust constraint (15). This will generate a protection on the original constraint, which is correlated with the protection determined for the aggregate constraint, and for which only a fraction Γ of the uncertainty budget is applied. The corresponding modified robust formulation considers the following constraints instead of (12):

$$\sum_{v} x_{jt2v} - \overline{P}_{j2} u_{jt2} + \upsilon_t \Gamma + \alpha_{jt} \leqslant 0 \quad \forall t, \ \forall j \in J_2,$$
$$\sum_{v} \sum_{j \in J_2} x_{jt2v} - \sum_{j \in J_2} \overline{P}_{j2} u_{jt2} + \upsilon_t \Gamma \sum_{j \in J_2} F_{jt} + \sum_{j \in J_2} \alpha_{jt} \leqslant 0 \quad \forall t, \tag{17}$$
$$\alpha_{jt} + \upsilon_t \geqslant \delta \overline{P}_{j2} \beta_{jt} \quad \forall t, \ \forall j \in J_2,$$
$$u_{jt2} \leqslant \beta_{tj} \quad \forall t, \ \forall j \in J_2.$$

The advantage of this set of constraints is that it allows controlling simultaneously the variation of the parameters P_{j2}, while the original set of constraints (12) by themselves, with only one parameter per constraint, collapses to the worst case scenario.

3.2. Adversarial approach

As explained in Section 2, the adversarial formulation allows us to obtain the "correct" robust solution without explicitly constructing a robust counterpart of the problem. For the particular case of constraints (12) we have that the protected constraints are:

$$\sum_{v} x_{jt2v} + \gamma_{jt}(x, u, \Gamma) \leqslant \overline{P}_{j2} u_{jt2}, \quad \forall j \in J_2, \ \forall t, \tag{18}$$

where $\gamma_{jt}(x, u, \Gamma)$ are the protection functions for each constraint. As we want the uncertain data to be in the set defined by (13), this implies that the adversarial problem will have the following form:
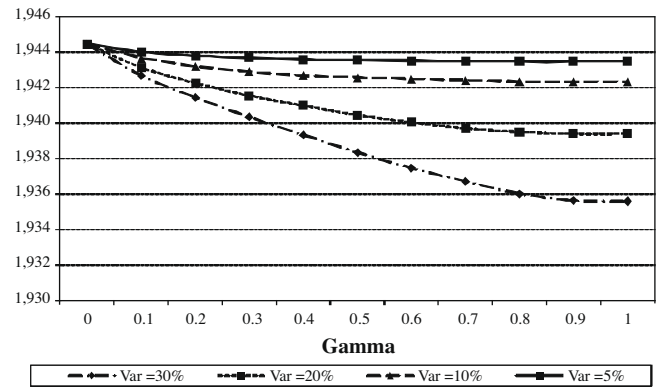

Fig. 1. Profit for different variabilities and uncertainty budgets.

$$\text{Max} \quad \sum_{t \in T} \sum_{j \in J_2} \gamma_{tj}$$
$$\text{s.t.} \quad \gamma_{tj} \leqslant z_{tj} u_{jt2} \overline{P}_{j2} \delta \quad \forall t, \ \forall j \in J_2,$$
$$\sum_{j \in J_2} z_{tj} \leqslant \Gamma \sum_{j \in J_2} F_{jt} \quad \forall t, \tag{19}$$
$$0 \leqslant z_{tj} \leqslant 1 \quad \forall t, \ \forall j \in J_2,$$
$$\gamma_{tj} \geqslant 0 \quad \forall t, \ \forall j \in J_2.$$

Hence, the adversarial iteration will compute an optimal solution for the original problem. That solution, specifically the u_{jt2} values (corresponding to workers assigned), will be introduced as a parameter in (19). This problem will be solved and the values z_{tj} will be used to construct a new scenario of productivity parameters to generate new constraints, which will be added to the formulation. Notice that, contrary to the general case explained in Section 2, this adversarial problem links all the constraints (12) for each period of time. This allows us to explicitly represent the fact that not all the productivity parameters in the various constraints will achieve their worst case simultaneously, depending on the specific value of Γ we may use.

4. Computational results

The models described in the previous section were tested on the base case data set used by Ferrer et al. [14], which consists of 20 blocks with a planning horizon of 18 days. For the labor productivity uncertainty, we considered four levels of variability with respect to the nominal value: 5%, 10%, 20%, and 30%, meaning that we used $\delta = 0.05$, 0.10, 0.20, and 0.30. The value of Γ was normal-
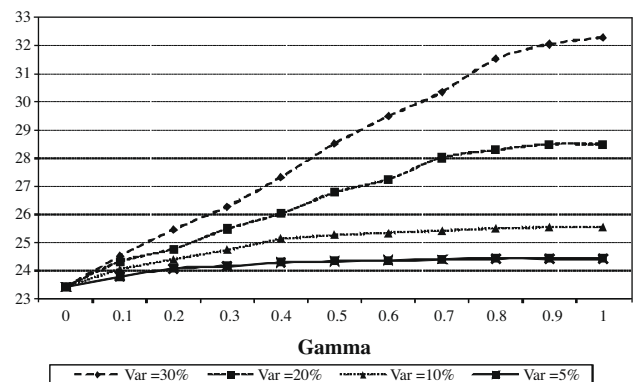

Fig. 2. Evolution of the operational cost for different variabilities and uncertainty budgets.
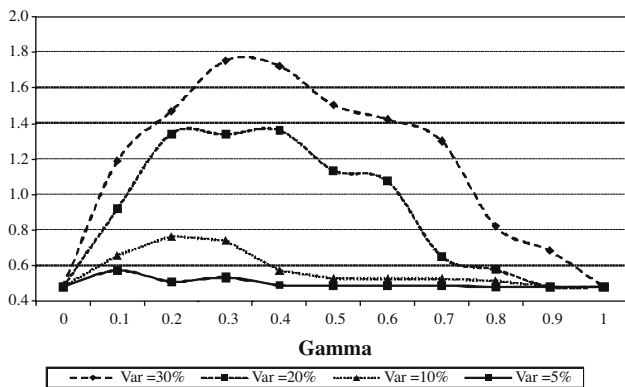
Fig. 3. Evolution of the quality cost for different variabilities and uncertainty budget.

Fig. 5. Percentage of infeasibleness assuming a uniform distribution of the uncertain parameter.

ized to vary from 0 to 1 in steps of 0.1. Recall that a value of Γ of 1 is equivalent to the worst case scenario.

The harvest planning model generates a problem with 6,163 constraints and 2452 variables, of which 1415 are integer. This is a difficult problem to solve to optimality using branch and bound. Nevertheless, we obtained solutions within 0.1% of the optimum value in less than 2 minutes using a standard Pentium 4 PC. The modifications to the model to make it more robust had little impact on the computational performance.

In the adversarial approach, the iterations were coded in AMPL and solved using GLPK. The stopping criteria was a decrease in less than 10^{-6} in the objective value, which is less than 0.5% of the original quality cost, and less than 0.01% of the original operations cost. The number of adversarial iterations was 7 to 8, for high values of Γ and δ, and between 4 and 5 in the best cases.

4.1. Aggregated modified robust model

Fig. 1 summarizes the objective values obtained by solving the modified robust model with different variabilities (δ) and uncertainty budgets (Γ). The graph clearly shows how the optimal value (maximum profit) of the problem deteriorates as we increase the robustness of the solution, that is, as Γ increases. The graph also shows that this deterioration is greater for problems with larger variabilities. If we look more closely at two of the components of this objective, namely the operational cost and the quality deterioration cost, we can see that the operational costs also deteriorate as the robustness is increased (Fig. 2), and the quality costs are minimum for values of Γ of 0 and 1, and achieve a maximum at around 0.3 (Fig. 3). This is because, for low values of Γ,
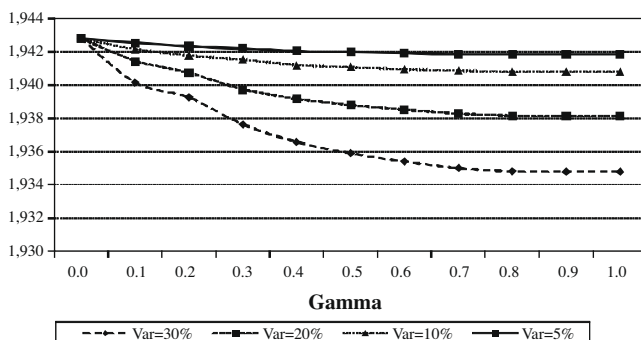
the productivity is only affected at the top quality days, and the model chooses to move harvesting operations to non optimal quality days, increasing the quality cost but lowering the operational cost by improving productivity. As Γ values approach 1, productivity is decreased in all feasible harvest days, so there is no operational benefit in moving the harvest operations to other days, going back to the original harvest schedule and lowering the quality cost.

4.2. Adversarial approach

Fig. 4 summarizes the objective values obtained by solving the adversarial approach model with different variabilities (δ) and uncertainty budgets (Γ). Although the overall behavior is quite similar to the one of the aggregated modified robust model, the quality cost component is much less sensitive to changes in the values of δ and Γ. The maximum variation of quality cost was less than 0.01%. This means that this model increases the use of labor to preserve the schedule, which results in a more conservative solution. It is up to the decision maker to decide which of the two aspects of the cost, operational or quality deterioration, is more important.

4.3. Feasibility analysis through Monte Carlo simulation

Both the aggregated modified robust model and the adversarial approach generate solutions with deteriorating objective values as the uncertainty (δ) and the robustness (Γ) increase. However, since we are not considering the worst case scenario, we also have to be

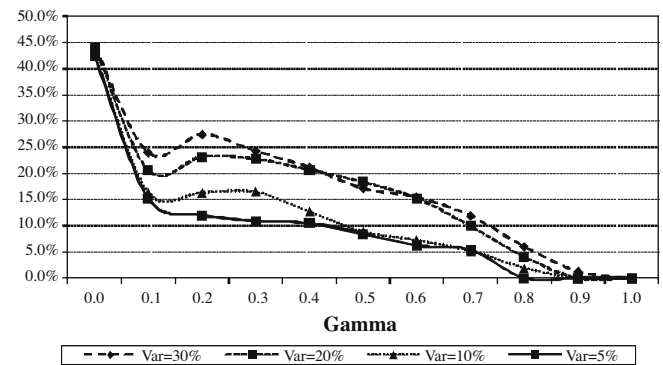**Fig. 4.** Objective values for different variabilities and uncertainty budgets using the adversarial approach model.
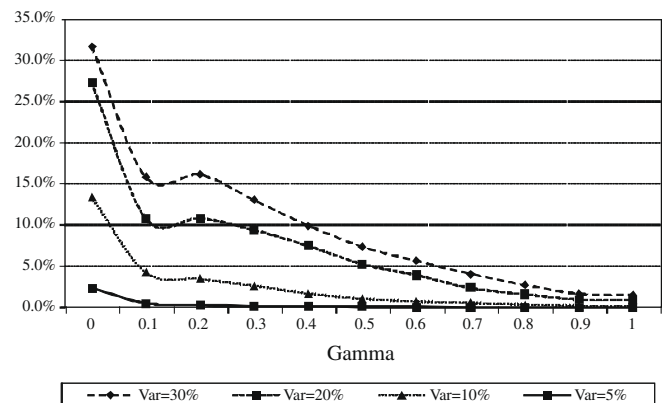

Fig. 6. Percentage of severe infeasibleness assuming a 95% coverage normal distribution.

Table 1
Percentage of infeasibleness (severe) for adversarial solutions assuming a 95% coverage normal distribution.

Γ	Var = 30%	Var = 20%	Var = 10%	Var = 5%
0.1	36% (26%)	36% (22%)	36% (12%)	37% (2%)
0.3	24% (18%)	23% (14%)	24% (7%)	24% (1%)
0.5	17% (12%)	17% (10%)	17% (5%)	16% (1%)
0.7	12% (9%)	12% (7%)	12% (3%)	12% (1%)
0.9	11% (8%)	11% (6%)	11% (3%)	11% (0%)

concerned with the feasibility of the solutions generated by the different models. To study this, we randomly generated 400 scenarios of the productivity parameters P_{j2}. We also assumed different distributions: a uniform distribution for each uncertain parameter, within the specified interval and level of uncertainty; a normal distribution within the interval, with a 95% coverage (which means the specified variability corresponded to 1.96 times the standard deviation of the distribution); and a normal distribution within the interval, with a 99.9997% coverage (equivalent to a 6-sigma coverage). We recorded the percentage of cases in which the solution was infeasible, and also the cases in which the solution was "severely infeasible", which we defined as those that violated the right hand side of the constraint by more than 5%.

4.3.1. Aggregated modified robust model

The results of the aggregated modified robust model using the different distributions were similar. Fig. 5 shows the percentage of infeasibilities obtained assuming a uniform distribution of the productivity parameters P_{j2}. Note that for a very low value of Γ, the percentage of infeasibilities is quite high, over 40%. However, it drops very quickly as Γ approaches 0.1. When variability is relatively high, the percentage of infeasibilities increases a little but then again starts to diminish until it is zero for a value of Γ of 1, which corresponds to the worst case situation. The graphs for the other distributions were all similar in shape.

Fig. 6 shows the percentage of severe infeasibleness obtained assuming a 95% coverage normal distribution of the productivity parameters P_{j2}. Again the graphs for the other distributions are similar. Note that the behavior for severe infeasibleness is similar to that of total infeasibleness, except that the values are lower and the differences between different levels of variability are larger.

4.3.2. Adversarial approach

The results for the adversarial approach are shown in Table 1, for the 95% coverage normal distribution. The results for the other distributions are similar. Note that the adversarial approach provides solutions which are more feasible than the aggregated modified robust model. However, we can also see that there is a significant difference when we consider total infeasibleness and severe infeasibleness. Table 1 shows that the percentage of feasible solutions, or almost feasible solutions (not severely infeasible), is around 95% for a value of Γ of 0.7, with a deterioration of the objective value of around 4%, depending on the assumed variability. Since in most practical applications, it is possible to tolerate some degree of infeasibility, this deterioration would probably be acceptable for many decision makers, in exchange for a more robust solution.

4.4. Effects on the Schedule

Besides the deterioration of the value of the optimal solution, and the possibility of obtaining infeasible solutions, it is also important to see how the robustness of a solution is reflected in the specific schedule generated by the model. Since showing a comparison of all the different schedules would take too much

Table 2
Comparison between the schedule generated using the non-robust model with those generated by the modified robust model.

Block	Nonrobust solution Obj. Fun.: 1,944				$\Gamma = 0.4$ Obj. Fun.: 1,939				$\Gamma = 0.9$ Obj. Fun.: 1,936			
	Day				Day				Day			
	7	8	9	10	7	8	9	10	7	8	9	10
a1	40	−	−	⊗	25	15	−	⊗	25	15	−	⊗
a2	−	85	−	−	25	15	−	45	25	60	−	−
a3	85	−	−	−	25	15	−	45	85	−	−	−
a4	27	13	−	⊗	25	15	−	⊗	25	15	−	⊗
a5	40	−	−	⊗	25	15	−	⊗	25	15	−	⊗
a6	⊗	−	82	−	⊗	−	38	45	⊗	−	82	−
a18	3	37	−	⊗	25	15	−	⊗	25	15	−	⊗
a19	40	−	−	⊗	25	15	−	⊗	25	15	−	⊗

space, we show in Table 2 a comparison between the schedule obtained using the basic model, without robustness, and the schedules generated using the aggregated modified robust model, with values of Γ of 0.4 and 0.9. The data shows the number of workers assigned to each block in a 4-day period, which had more activity, and only the blocks that were scheduled. The squares indicate the optimal date for harvest and the $\otimes$ symbol indicates that it is not feasible to harvest on that date. The full schedule was for 18 days and 20 blocks.

In the schedule with $\Gamma = 0.40$, harvesting is distributed in more days, and is less concentrated, with less daily workers. With $\Gamma = 0.90$, the schedule is again more concentrated and is similar to the non-robust one. It is important to remember that with $\Gamma = 0.40$ the quality cost was significantly higher, which is consistent with the lower percentage of the harvest on the optimal date of that schedule. Finally, with $\Gamma = 0.90$ the schedule tends to the worst case and the percentage of infeasibilities is close to 0.

5. Conclusions

Optimizing a problem, when faced with uncertainties, is difficult, especially when only partial knowledge of these uncertainties is available. We have presented an approach for doing this to support wine grape harvesting decisions in the wine industry.

The main uncertain parameter of the wine grape harvesting model we developed, labor productivity, actually reflects various sources of variability. We handled this uncertainty with a robust formulation that generates good feasible solutions. The proposed robust formulation departs from the traditional one by controlling the simultaneous variation of a set of parameters across different constraints instead of in the same constraints, as it is done in the standard theory of robust optimization. This was accomplished by adding an aggregate constraint to which we applied the robust reformulation, applying at the same time robust elements to the original constraints, in order to avoid the worst case and have a higher chance of getting feasible solutions. In that sense, our approach is heuristic but provides solutions which behave well in terms of expected feasibility and conservatism in the objective value. The feasibility of the solutions is studied using Monte Carlo simulation of various scenarios.

We also tested an adversarial formulation, as proposed by Bienstock and Ozbay [9], which has the advantage of producing a better representation of the true robust problem. This approach has the potential drawback, however, of having to iterate between a master problem and an adversarial subproblem that computes new scenarios. For larger instances of the problem, this might prove to be too time consuming.

For the purpose of comparison with the "correct" adversarial approach, the results obtained using the aggregated robust reformulation are very satisfactory, although there is a significant difference in the behavior of the generated schedule in terms of quality cost. This could be an important issue when applying the approach in some wineries. However, for the test case and a value of Γ of 0.7, we were able to obtain solutions with a feasibility probability close to 90%, estimated through Monte Carlo simulation.

The results show that the proposed approach is able to balance obtaining a feasible solution with objective value deterioration. However, if quality is a dominant factor, it might be better to have a model closer to the worst case situation.

In this paper, we have studied different ways to handle uncertainty in a wine grape harvesting problem. We focused on only one of the sources of uncertainty, namely labor productivity. Future extensions of the approach should allow analyzing other sources of uncertainties, like the processing capacity of the wineries. This last aspect is important as the actual capacity of the production facilities varies depending on the results of wine fermentation, a process which is essentially variable. One characteristic of this agricultural problem is that little or nothing is known about the exact pattern of variability in labor productivity. This is precisely the situation in which we believe a robust optimization formulation is best suited. Although a much more precise model might be built if we had distributional information, the fact that in the robust formulation we can change the degree of variability (either by changing the overall variation or the budget of uncertainty) allows at least to assess the sensitivity of the model to uncertainty in the data. This should prove to be very useful to decision makers.

Another aspect of the problem that we have not addressed is the possible correlation between labor productivity and other variables. For instance, productivity in certain harvesting blocks might follow a particular pattern over time. The current formulation implicitly assumes that all variations are independent. One possible way to address correlations is by mean of explicitly incorporating it into the adversarial problem (11), or by means of building an explicit robust counterpart that incorporates correlated data. This will be the subject of future research.

Acknowledgements

Appendix A. The harvest planning model

We describe here the model we are using for scheduling harvesting operations. An earlier version of this model is presented in [14]. The main decision variables are how much labor to assign to harvesting operations in each block of the vineyard. For each block, and depending on the kind of grapes, it is assumed that an estimate of the optimal harvesting date is available. This estimate is obtained by the oenologist based on laboratory samples and the evolution of the season. Besides this specified date, a time window is also determined in which harvesting should take place. There is also an estimate of the volume of grapes available in each block. This, together with an estimate of labor productivity can be used to determine how many workers are needed. The model incorporates cost considerations in the labor assignment but also it considers a cost for not harvesting in the optimal date. This means that we can use any day in the feasible time window, but we have to pay a price if we do not satisfy the optimal day. This quality penalty (a "quality loss function", in the sense of Tagushi and Clausing [25]), will be larger the better quality the grapes are. Normally grapes are used for different qualities of wine: premium, "reserve", "varietal" and bulk wine with no origin or variety denomination. The model also considers the cost of moving the harvesting operations from one block to another and balances this with the operational and quality cost. Other constraints are also taken into account, like labor availability and winery processing capacity. The results in [14] show good response of the model when scheduling operations. Results for a fictitious case as well as for a real data are presented in that paper.

In this paper, we have made some small modifications to the original model, in particular we have simplified some of the labor constraints. We describe now the specific model used. The main decision is which blocks to harvest and when to do it. This translates into a decision variable indicating the volume of grapes (in kilogram) harvested from a given block in a given day, with the additional qualification of the destination winery and harvesting method (by hand or using machinery). This decision is affected by the estimated optimal harvest day and constrained by a feasibility window. The quality loss, as explained earlier, is translated into a specific cost in the objective function associated to this variable. The formulation considers that the blocks can be harvested either by machine or by hand and the model has to compute how many hours of labor are needed in each case. It is assumed that a block can be assigned to a single winery only. Also, a minimum harvest activity in a given block is required if we decide to intervene the block. Additional variables and constraints are included to represent the routing of the harvesting operations from one block to another. This part leads to a kind of traveling salesman problem formulation for which the MTZ [21] formulation is used. The description of the model is as follows:

A.1. Sets

K	harvesting methods, $k = 1$ for mechanical harvest and $k = 2$ for manual harvest.
R	grape types, $r = 1$ for premium, $r = 2$ for reserve, $r = 3$ for varietal, and $r = 4$ for bulk.
J_k	blocks that have some grapes available for harvest method $k \in K$.
$J = J_1 \cup J_2$	total blocks.
J_r	blocks with grapes type $r \in R$.

A.2. Parameters

D_{ij}	distance between blocks i and j.
Q_{jt}	quality loss factor for block j at day t.
C^H	workers hiring cost.
C^F	workers firing cost.
C_k^U	unit cost of productive resource k. For $k = 1$ it is cost per machine hour and for $k = 2$ it is cost per worker.
P_{jk}	productivity of harvest method k at block j, in kilogram/machine-hour, for $k = 1$ or kg./worker, for $k = 2$.
G_{jkv}	estimated kilograms of grapes available at block j for harvesting method k and winery v.
F_{jt}	feasibility indicator for block j at day t (it is 1 if day t is in the time window).
L_{tkv}	estimated winery capacity for day t, winery v, and method k.

B_{jtk} maximum volume that can be harvested in block j at day t using harvesting mode k.

M a "big M" parameter.

V_k minimum daily harvest for a single block, using harvest method k.

A_t Machine availability, in hours, in period t.

N minimum number of workers that can be used to harvest a block using harvest method 2.

S_r Price of grape type r.

C^R relocating operations cost.

A.3. Variables

x_{jtkv} kilograms harvested at block j, day t, harvest method k, and destination winery v.

y_{jtkv} binary variable to indicate if it is possible to harvest block j, at time t, using method k, for winery v.

v_{jtk} binary variable to indicate whether block j is harvested at time t using method k.

z_{ijtv} binary variable. It is 1 if there is a direct route between i and j, at time t for winery v.

τ_{jtv} variable for eliminating subtours in the TSP.

w_t^H number of workers hired at day t.

w_t^F number of workers fired at day t.

u_{jtk} Productive resources used in block j at period t. If $k = 1$ it is machine hours if $k = 2$ it is number of workers.

A.4. Mathematical model

$$\text{Max}: \quad \sum_t \left(\sum_r S_r \sum_{j \in J_r}(1 - Q_{jt}) \sum_v \sum_k x_{jtkv} - \sum_k \sum_{j \in J_k} C_k^U u_{jtk} \right.$$

$$\left. - C^H w_t^H - C^F w_t^F - C^R \sum_v \sum_{i,j/i \neq j} D_{ij} z_{ijtv} \right)$$

s.t.

$$\sum_j x_{jtkv} \leqslant L_{tkv} \quad \forall t, k, v, \tag{A.1}$$

$$\sum_t x_{jtkv} \leqslant G_{jkv} \quad \forall j, k, v, \tag{A.2}$$

$$x_{jtkv} \leqslant L_{tkv} y_{jtkv} \quad \forall j, t, k, v, \tag{A.3}$$

$$x_{jtkv} \geqslant V_k y_{jtkv} \quad \forall j, t, k, v, \tag{A.4}$$

$$\sum_v y_{jtkv} \leqslant M v_{jtk} \quad \forall j, t, k, \tag{A.5}$$

$$v_{jtk} \leqslant \sum_v \sum_{s:s \leqslant t} y_{jskv} \quad \forall j, t : F_{jt} = 1 \forall k, \tag{A.6}$$

$$y_{jtkv} \leqslant \sum_{s:s \leqslant t} v_{jsk} \quad \forall j, t : F_{jt} = 1 \ \forall k, v, \tag{A.7}$$

$$\sum_t \sum_k v_{jtk} \geqslant 1 \quad \forall j : G_{jkv} \neq 0, \tag{A.8}$$

$$y_{jtkv} \leqslant F_{jt} \quad \forall j, t, k, v, \tag{A.9}$$

$$\sum_v x_{jtkv} \leqslant P_{jk} u_{jtk} \quad \forall j, t, k, \tag{A.10}$$

$$\sum_{j \in J_2} u_{jt2} = \sum_{j \in J_2} u_{j(t-1)2} + w_t^H - w_t^F \quad \forall t > 1, \tag{A.11}$$

$$u_{jt2} \geqslant N v_{jt2} \quad \forall j, t \in J_2, \tag{A.12}$$

$$u_{jtk} \leqslant M \sum_v y_{jtkv} \quad \forall t, k, j \in J, \tag{A.13}$$

$$\sum_{j \in J_1} u_{jt1} \leqslant A_t \quad \forall t, \tag{A.14}$$

$$\tau_{jtv} \geqslant \tau_{itv} + 1 - (\text{card}(J) - 1)(1 - z_{ijtv}) \quad \forall i, j, t, v : i \neq j, \tag{A.15}$$

$$\sum_{j:j \neq i} z_{ijtv} = y_{it2v} \quad \forall i, t, v : F_{it} \neq 0, \tag{A.16}$$

$$\sum_{i:i \neq j} z_{jitv} = y_{jt2v} \quad \forall j, t, v : F_{jt} \neq 0, \tag{A.17}$$

$$z_{tijv}, y_{jtkv}, v_{jtk} \in \{0, 1\},$$
$$\tau_{jtv} \in \mathbb{Z}_0^+,$$
$$w_t^H, w_t^F, u_{jtk}, x_{jtkv} \geqslant 0.$$

In this formulation, the constraints (A.1)–(A.9) are basically capacity constraints and the definition constraints for the binary variables. (A.10) establishes the productivity requirement while (A.11) defines the evolution of labor. (A.12) and (A.13) define the minimum harvesting requirement, and (A.13)–(A.15) are various capacity and labor availability constraints. (A.16) and (A.17) are used for the routing part of the model and originate on the MTZ formulation for the traveling salesman problem. The first two lines of the objective function correspond to the quality cost, expressed as a penalty on the price of the grapes. The third and fourth lines correspond to the operational cost, including labor and translation costs.

References

[1] O. Ahumada, J. Villalobos, Application of planning models in the agri-food supply chain: A review, European Journal of Operational Research 195 (2009) 1–20.

[2] K. Al-Qahtani, A. Elkamel, Multisite facility network integration design and coordination: An application to the refining industry, Computers & Chemical Engineering 32 (10) (2008) 2189–2202.

[3] F. Allen, E. Schuster, Controlling the risk for an agricultural harvest, Manufacturing & Service Operations Management 6 (3) (2004) 225–236.

[4] A. Ben-Tal, A. Nemirovski, Robust convex optimization, Mathematics of Operations Research 23 (1998) 769–805.

[5] A. Ben-Tal, A. Nemirovski, Robust solutions of uncertain linear programs, Operations Research Letters 25 (1999) 1–13.

[6] A. Ben-Tal, A. Nemirovski, Robust solutions of linear programming problems contaminated with uncertain data, Mathematical Programming 88 (2000) 411–424.

[7] D. Bertsimas, M. Sim, Robust discrete optimization and network flows, Mathematical Programming 98 (2003) 43–71.

[8] D. Bertsimas, M. Sim, The price of robustness, Operations Research 52 (1) (2003) 35–53.

[9] D. Bienstock, N. Ozbay, Computing robust basestock levels, Discrete Optimization 5 (2) (2008) 389–414.

[10] J. Birge, F. Louveaux, Introduction to Stochastic Programming, Springer-Verlag, New York, 1997.

[11] L. Contesse, J. Ferrer, S. Maturana, A mixed-integer programming model for gas purchase and transportation, Annals of Operations Research 139 (1) (2005) 39–63.

[12] L. El Ghaoui, H. Lebret, Robust solutions to least-square problems to uncertain data matrices, SIAM Journal on Matrix Analysis and Applications 18 (4) (1997) 1035–1064.

[13] L. El Ghaoui, F. Oustry, H. Lebret, Robust solutions to uncertain semidefinite programs, SIAM Journal on Optimization 9 (1) (1998) 33–52.

[14] J.-C. Ferrer, A. Mac Cawley, S. Maturana, S. Toloza, J. Vera, An optimization approach for scheduling wine grape harvest operations, International Journal of Production Economics 112 (2) (2008) 985–999.

[15] C. Floudas, X. Lin, Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications, Annals of Operations Research 139 (1) (2005) 131–162.

[16] N. Islam, M. Quaddus, An Appraisal of uncertainties in the western australian wine industry supply chain, in: 49th Annual Conference of the Australian Agricultural and Resource Economics Society, Coffs Harbour, New South Wales, Australia, 2005.

[17] S. Karabuk, Production planning under uncertainty in textile manufacturing, Journal of the Operational Research Society 59 (4) (2008) 510–520.

[18] B. Kazaz, Production planning under yield and demand uncertainty with yield-dependent cost and price, Manufacturing & Service Operations Management 6 (3) (2004) 209–224.

[19] P. Kouvelis, G. Yu, Robust Discrete Optimization and its Applications, Kluwer Academic Publishers, Dordecht, The Netherlands, 1997.

[20] J. Meyers, J. Miles, J. Faucett, F. Fathallah, I. Janowitz, R. Smith, Smaller loads reduce risk of back injuries during wine grape harvest, California Agriculture 60 (1) (2006) 25–31.

[21] C.E. Miller, A.W. Tucker, R.A. Zemlin, Integer programming formulation of traveling salesman problems, Journal of ACM 7 (4) (1960) 326–329.

[22] J. Mulvey, R. Vanderbei, S. Zenios, Robust optimization of large-scale systems, Operations Research 43 (2) (1995) 264–281.

[23] Y. Nesterov, A. Nemirovski, Interior-Point Polynomial Algorithms in Convex Programming, SIAM Studies in Applied Mathematics, Philadelphia, 1994.

[24] A. Soyster, Convex programming with set-inclusive constraints and applications to inexact linear programming, Operations Research 21 (1973) 1154–1157.

[25] G. Tagushi, D. Clausing, Robust quality, Harvard Business Review 68 (1) (1990) 65–75.

[26] C.-S. Yu, H.-L. Li, A robust optimization model for stochastic logistic problems, International Journal of Production Economics 64 (2000) 385–397.