Stochastic Quadratic Knapsack with recourse

Lisser Abdel* Lopez Rafael* Hu Xu*

*Laboratoire de Recherche en Informatique, Universite Paris-Sud XI
Bat. 490, Universit Paris-Sud, 91405 Orsay France

Abstract

This paper is dedicated to a study of different extensions of the classical knapsack problem to the case when different elements of the problem formulation are subject to a degree of uncertainty described by random variables. This brings the knapsack problem into the realm of stochastic programming. Four different model formulations are proposed, based on the introduction of probability constraints. The first one is a two-stage quadratic knapsack with recourse (2QKR), and serves as the base on which we develop the three other models. The second one is a 2QKR with a probability constraint on the capacity of the knapsack on the first stage. The third one is a 2QKR where we introduce a probability constraint on the capacity of the knapsack in the second stage. Finally, the last model is also a 2QKR which uses probability constraints on the capacity constraints of both stages. As far as we know, this is the first time such a constraint has been used in a two-stage model. The solution techniques are based on the semidefinite relaxations. This allows for solving large instances, for which exact methods cannot be used.

Keywords: *Semidefinite programming, knapsack, stochastic, recourse.*

1 Introduction

The knapsack problem (KP) is a well-known and well-studied problem in combinatorial optimization. Knapsack problems are often used to model industrial situation, financial decisions or network design problems. They may also appear as sub-problems of larger or more complex problems. The most famous form of KP is the single constraint binary version: we are given N items, with return p_i for the item i, and a weight w_i for the item i, with $i = 1, ..., N$, and a knapsack capacity c. The problem is to select a subset of items so that the weight of the subset does not exceed c, and gives a maximal total return. In this form, the problem is known to be NP-Hard [GJ79], and has been intensively studied in the past decades, and several exact and approximate algorithms are known for this problem. In particular, it admits a FPTAS [IK75]. For the quadratic knapsack problem, a survey done by David Pisinger [Pis07] gives detailed information on the problem and a number of results on the performance of various relaxations and algorithms used to solve or approximate the problem.

In the case of modeling financial decisions, transportation, or production plans, however, this formulation shows its limitations, since it does not take into account uncertainty on the problem parameters, such as the prices p_i or the weights w_i. Similarly, such decisions are not static, and this model cannot take into account new information available on the prices or the weights. There have been several studies done on the stochastic knapsack problem in the past years: work has been done to find heuristics [CB98], approximation algorithms [KP98], [DGV04], [SS06]...

Stochastic knapsack problems can reach a number of binary variables and constraints of such magnitude that commercial packages cannot find a solution in a reasonable time or memory space, requiring the use of linear relaxations to find an upper bound on the problem. While linear relaxations were successful for many combinatorial optimization problems, it turns out that knapsack problems, especially their quadratic formulation, can not be approximated tightly by linearization based methods. Stronger relaxation methods, namely semidefinite relaxations (called SDP thereafter), have turned out to be particularly interesting for such combinatorial optimization problems [GW95], [HR98]. See also [Pis07] for a survey on the quadratic knapsack problem.

More precisely, semidefinite programming is a recent development of convex optimization, which deals with optimization problems over symmetric positive semidefinite matrices with linear cost function and linear constraints. Groetschel and al. showed that semidefinite optimization problems can be solved in polynomial time [GLS88].

In this paper we present the two-stage quadratic knapsack with recourse that is used as the base for three variants of stochastic optimization problems : the first one is a two-stage quadratic knapsack with probability constraint on the capacity in the first stage. The probability constraint is used to model the risk we are willing to take when making our initial decision. We only know some information about the weights of the items, but we have to take a decision with this limited knowledge at the risk of breaking the capacity constraint with the knowledge that a second stage decision (the recourse) will come after, allowing to correct the decision to some extent. In this paper, we consider a two-fold recourse, in the sense that items can be removed if they turn out to be suboptimal or if their weight appears to be too heavy, or added if they appeared to be uninteresting at first, but reveal to be desirable. The second model is a two-stage quadratic knapsack with probability constraint on the second stage capacity constraint. In this model, we also have a two-stage formulation which models a situation where we make a decision with limited information, but where the second stage decision is made after receiving additional (but still incomplete) information about the weight, or prices. This allows to modify initial decisions. In this model, since at the second step, some information is still unknown, we face a risk of breaking the capacity constraint when taking the second stage decision. Finally, the last model combines the probability constraints on the capacity constraint in both stages. In this model, we are willing to take a risk in all the decisions.

This paper is organized as follows: first, we present the static knapsack problem with probability constraint wherein we give a presentation of the problem. Then we reformulate this problem under the form of an SDP problem and detail two different relaxations. We then present the stochastic knapsack problem with recourse. Finally, concluding remarks are given.

2 Two-stage quadratic knapsack problem

The initial decision is made during the first stage before knowing the realizations of the random variables. Then these realizations are (partially) revealed and the second stage decision is made which corrects the first stage decision, taking into account this information. This modeling scheme is known as *stochastic program with recourse*. Moreover, we introduce probability constraints in the second stage. To the best of our knowledge, this formulation has not been studied in the literature. We start by formulating the general quadratic stochastic program with recourse with probability constraint in the second stage. A generic quadratic stochastic problem with recourse can be modeled as follows:

$$\max_{x\in\{0,1\}} x^T C x + \mathbb{E}_\omega Q(u,\omega) \tag{1}$$

$$Rx \leq s \tag{2}$$

where (2) models generic linear constraints, with $R \in \mathbb{R}^{m\times n}$ and $s \in \mathbb{R}^m$. The second stage value is given by the solution of the problem:

$$Q(u,\omega) = \max_u u^T D(\omega) u \tag{3}$$

$$W(\omega)u + T(\omega)x \leq h(\omega) \tag{4}$$

In this model, the uncertainty is described by the probability vector ω with a given probability distribution. When the initial decision is made, ω is unknown. After this decision, part of the information is revealed. This corresponds to the realization of random vector ω. The second stage decision can then be taken, with knowledge of ω and of the first stage decision x. More details can be found in [AG08].

First stage decision

We now need to adapt this generic model to our knapsack problem. We assume we have n items, and each item is characterized by its value c_{ii}, and weight w_i, $i = 1 : n$. Each item pair is characterized in the same manner by its value c_{ij}. The objective is to maximize the value of the items contained in the knapsack, with the constraint that it has a limited capacity d. The selection of an item during the first stage is defined by a binary decision variable x_i which takes value 1 if the item i is included in the selection and 0 otherwise. The formulation of this first stage decision of the problem is the following:

$$\max_x \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_i x_j + \mathbb{E}_\omega Q(u, \omega) \tag{5}$$

$$\sum_{i=1}^{N} w_i x_i \leq d \tag{6}$$

The constraint (6) describes the knapsack capacity constraint. Equation (5) consists of two parts: the value of the knapsack during the first stage, and the expected value of the same knapsack during the second stage. This expected value depends on the items selected in the first stage (the vector x), and the realization of the random vector ω.

Second stage decision

After the first stage decision is made, the values of items may change, as well as their weight. During the second stage, the item i has the value $b_{ii}(\omega)$ and the weight $v_i(\omega)$. Each item pair (i, j) has the value $b_{ij}(\omega)$. Similarly to the first stage, there is a constraint on the capacity of the knapsack, which is subject to change too. We note the new capacity $h(\omega)$. The realization of vector ω is known before making the second stage decision.

The second stage decision allows to change the initial decision in order to correct mistakes which appear after extra information is known. There are two possibilities: first, an item which was selected during the first stage can be removed. In this case, we describe this decision by a binary variable u_i^- set to 1 if item i is removed from the knapsack, 0 otherwise. Likewise, an item that was previously rejected can be selected. In this case, we use a binary variable u_i, set to 1 if we select the item i in the second stage, 0 otherwise. Note that if item i was selected during the first stage, then if it is not removed during the second, it is considered selected again (i.e. $u_i = 1$) in the second stage. When an item i is removed, it comes at a cost, which includes penalties, such as time or manipulation costs necessary to reorganize the knapsack. This allows us to formulate the second stage decision as follows:

$$Q(u, \omega) = \max_{u, u^-} \sum_{i=1}^{N} \sum_{j=1}^{N} b_{ij}(\omega) u_i u_j - \sum_{i=1}^{N} \sum_{j=1}^{N} b_{ij}^-(\omega) u_i^- u_j^- \tag{7}$$

$$u_i \geq x_i - u_i^-, \ i = 1 : n \tag{8}$$

$$u_i^- \leq x_i, \ i = 1 : n \tag{9}$$

$$\sum_{i=1}^{N} v_i(\omega)(u_i + x_i - u_i^-) \leq h(\omega) \tag{10}$$

Equation (7) is our objective: we want to maximize the value of the knapsack where we deduce the cost of removing items previously selected. The two constraints (8) and (9) link the first stage and second stage decisions: the first one means that if an item i was selected during the first stage and not deselected, then it is necessarily considered selected during the second stage. Conversely, the constraint (9) means that only an item that was selected during the first phase can be deselected. Constraint (10) represents the capacity constraint in the second stage.

This model serves as the base on which we build stochastic extensions: first we introduce a probability constraint on the first stage of the problem, in the first sub-section, then in the second sub-section we look at the model where the probability constraint is on the second stage. Finally, in the third sub-section, we combine probability constraints in both stages.

3 Probability constraints in the two-stage quadratic knapsack

In order to model the risk, we need to introduce probability constraints. Taking a risk in a two-stage decision process can happen in either or both of the stages, therefore leading to three variants of the two-stage quadratic knapsack.

Two-stage quadratic knapsack problem with probability constraint in the first stage

In order to model risk-taking in the first stage decision, we have to replace the capacity constraint (6) with a probability constraint.

$$\mathbb{P}\left\{\sum_{i=1}^{N} w_i(\phi)x_i \leq d\right\} \geq (1-\alpha_1) \tag{11}$$

where ϕ is a probability vector with a given known distribution representing the uncertainty on the weights of the items, and α_1 is the risk we take of ignoring the capacity constraint. This case corresponds to a situation in which we must make a first decision under very limited knowledge (on the weights and on the future), knowing that we will be able to correct the decision in a second stage.

Two-stage quadratic knapsack problem with probability constraint in the second stage

Similarly to the case with a probability constraint in the first stage, the base problem (5)-(10) can be modified to allow risk-taking in the second stage decision. We have to replace constraint (10). This time, we model the added uncertainty on a second probability vector ψ, conditioned on the vector ω. While the realization of vector ω is known before making the second stage decision, only the distribution of ψ is known, conditioned on ω.

$$\mathbb{P}\left\{\sum_{i=1}^{N} v_i(\omega,\psi)(u_i + x_i - u_i^-) \leq h(\omega,\psi)|\omega\right\} \geq (1-\alpha_2) \tag{12}$$

This situation corresponds to the case where information about the future is very limited. In fact, the distribution of weights in the second stage is not known until after ω is realized.

Two-stage quadratic knapsack problem with probability constraint in both stages

Finally, the last variant combines risk taking in both stages. This combines the limited information of both cases above: the first decision is taken with uncertainty about the first stage weights, then information is revealed, but is still incomplete, which leads to a second stage decision under uncertainty. In this case the problem becomes:

$$\max_x \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_ix_j + \mathbb{E}_\omega Q(u,\omega) \tag{13}$$

$$\mathbb{P}\left\{\sum_{i=1}^{N} w_i(\phi)x_i \leq d\right\} \geq (1-\alpha_1) \tag{14}$$

$$Q(u,\omega) = \max_{u,u^-} \sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij}(\omega)u_iu_j - \sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij}^-(\omega)u_i^-u_j^- \tag{15}$$

$$u_i \geq x_i - u_i^-,\ i=1:n \tag{16}$$

$$u_i^- \leq x_i,\ i=1:n \tag{17}$$

$$\mathbb{P}\left\{\sum_{i=1}^{N} v_i(\omega,\psi)(u_i + x_i - u_i^-) \leq h(\omega,\psi)|\omega\right\} \geq (1-\alpha_2) \tag{18}$$

As we showed in [AG08], formulation (5)-(10) of the two-stage quadratic knapsack problem is quite general and covers many specific cases. Our formulation (5)-(10) is general enough to allow the modeling of additional characteristics which do not appear explicitly in the problem description. The main such feature is that this model allows a different composition of the allowable set of items during the first and the second stage. Since the weights of the items and the capacity of the knapsack may change (based on the realization of ω) between the first and second stage, it allows us to model cases where some items may not be allowed during either stage. For example, an item could be unavailable during the first stage and only available during the second stage, in which case it would have $w_i > d$. This also allows having different set of items for the two stages, which is traditional in recourse models: items which can only be decided on in the first phase will have a deselection cost b_{ii}^- set to a very large number (to prevent deselection), and a second stage value $b_{ii}, b_{ij} \leq 0$ (which means selection in the second phase is suboptimal, since it only affects the weight and does not increase the profit).

We will now work with this problem and reformulate it in order to be able to use resolution techniques like SDP. The first step of the reformulation is to rewrite the problem into deterministic equivalent problems. Then we will be able to use semidefinite relaxations on the deterministic equivalent problems.

4 Deterministic equivalent problem

In order to rewrite the stochastic quadratic knapsack with recourse into a deterministic form, we need to consider the case when the distribution of the random vectors ϕ, ω and ψ is concentrated in a finite number of points. We assume that the random vector ω is concentrated in the finite number of points $\omega_k, k = 1 : K$ with probabilities p_k^ω. We will refer to these points as *scenarios.* In this case the problem (5)-(10) can be rewritten as follows. The objective function in the first stage becomes:

$$\max_x \left[\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_ix_j + \sum_{k=1}^{K} p_k^\omega Q(u,k)\right] \tag{19}$$

with the second stage decision becoming:

$$Q(u,k) = \max_{u,u^-} \left[\sum_{i=1}^{N}\sum_{j=1}^{N} b_{ijk}u_{ik}u_{jk} - \sum_{i=1}^{N}\sum_{j=1}^{N} b_{ijk}^-u_{ik}^-u_{jk}^-\right] \tag{20}$$

The second stage constraint capacity also changes:

$$\sum_{i=1}^{N} v_{ik}(u_{ik} + x_i - u_{ik}^-) \leq h_k \qquad \forall k = 1:K \tag{21}$$

where

$$Q(u,\omega_k)=Q(u,k),\ b_{ij}(\omega_k)=b_{ijk},\ b^-_{ij}(\omega_k)=b^-_{ijk},\ v_i(\omega_k)=v_{ik},\ h(\omega_k)=h_k.$$

Substituting (20) into (19) and collecting constraints for each scenario we obtain an equivalent problem:

$$\max_{x,u_{ik},u^-_{ik}} \left[\sum_{i=1}^N\sum_{j=1}^N c_{ij}x_ix_j+\sum_{k=1}^K p^\omega_k\left(\sum_{i=1}^N\sum_{j=1}^N b_{ijk}u_{ik}u_{jk}-\sum_{i=1}^N\sum_{j=1}^N b^-_{ijk}u^-_{ik}u^-_{jk}\right)\right] \tag{22}$$

$$\sum_{i=1}^N w_ix_i\le d \tag{23}$$

$$u_{ik}\ge x_i-u^-_{ik},\ i=1:n,\ k=1:K \tag{24}$$

$$u^-_{ik}\le x_i,\ i=1:n,\ k=1:K \tag{25}$$

$$\sum_{i=1}^N v_{ik}(u_{ik}+x_i-u^-_{ik})\le h_k\ ,\ k=1:K \tag{26}$$

Probabilistic constraints

Probability constraints (11) and (12) are reformulated as deterministic equivalent constraints. Suppose that the random vector ϕ (resp. ψ) is concentrated in the finite number of points $\phi_l, l=1:L$ (resp. $\psi_{kr}, k=1:k, r=1:R$) with probabilities p^ϕ_l (resp. p^ψ_{kr}) such that

$$\sum_{l=1}^L p^\phi_l=1,\ \ p^\phi_l\ge 0$$

$$\sum_{r=1}^R p^\psi_{kr}=1,\ \ p^\psi_{kr}\ge 0,\ k=1:K$$

Then constraint (11) is equivalent to the pair:

$$\begin{cases}\displaystyle\sum_{i=1}^N w_{il}x_i\le d_l,\ l\in\Gamma\\ \displaystyle\sum_{l\in\Gamma} p_{kr}\ge 1-\alpha_1,\ k=1:K\end{cases}$$

where $w_{il}=w_i(\phi)$, $d_l=d(\phi)$ and Γ is a subset of scenarios of set $\{1,...,L\}$ where the capacity constraint is satisfied, while the set $\{1,...,L\}\setminus\Gamma$ corresponds to the scenarios where risk is taken. These constraints can be reformulated as binary constraints by introducing the auxiliary binary variable y^ϕ_l (resp. y^ψ_{kr}) for each scenario $l=1:L$ (resp. observation $r=1:R$ and scenario $k=1:K$) as follows:

$$y^\phi_l=\begin{cases}0 & \text{if} \quad l\in\Gamma\\ 1 & \text{otherwise}\end{cases}$$

This yields the following deterministic equivalent constraints:

$$\begin{cases}\displaystyle\sum_{i=1}^N w_{il}x_i\le d_l+M^\phi_l y^\phi_l\\ \displaystyle\sum_{l=1}^L p^\phi_l y^\phi_l\le\alpha_1\end{cases} \tag{27}$$

where M_l^ϕ is an arbitrary number such that

$$M_l^\phi \geq \sum_{i=1}^{N} w_{il} - d_l$$

In the same manner, constraint (12) is equivalent to:

$$\begin{cases} \sum_{i=1}^{N} v_{ikr}(u_{ik} + x_i - u_{ik}^-) \leq h_{kr}, \ r \in \Lambda_k, \ k = 1:K \\ \sum_{r \in \Lambda_k} p_{kr} \geq 1 - \alpha_2, \ k = 1:K \end{cases}$$

where $v_{ikr} = v_{ik}(\psi_{kr})$, $h_{kr} = h_k(\psi_{kr})$ Λ_k is a subset of $\{1, ..., R\}$. Again we can reformulate them using binary variables:

$$y_{kr}^\psi = \begin{cases} 0 & \text{if} \quad r \in \Lambda_k \\ 1 & \text{otherwise} \end{cases}$$

yielding the following deterministic equivalent constraints:

$$\begin{cases} \sum_{i=1}^{N} v_{ikr}(u_{ik} + x_i - u_{ik}^-) \leq h_{kr} + M_k^\psi y_{kr}^\psi, \ r = 1:R, \ k = 1:K \\ \sum_{r=1}^{R} p_{kr}^\psi y_{kr}^\psi \leq \alpha_2, \ k = 1:K \end{cases} \tag{28}$$

where M_k^ψ is an arbitrary number such that

$$M_k^\psi \geq \max_r \sum_{i=1}^{N} v_{ikr} - h_{kr}$$

5 Conclusion

In this paper we detailed the models for two-stage quadratic knapsack problems with recourse, on which we introduced probability constraints on the first stage, second stage, or both. Currently, numerical experiments are underway to complete those presented in [AG08]. Furthermore, these experiments add a distinction between models where the deselection of items is allowed – like presented in this paper – and the case where items can only be added, and not removed. These experiments compare the problem size and result between a MIP formulation solved by CPLEX, a LP relaxation, and various SDP relaxations for various sizes of instances of each variation. In particular, we show that these problems, even for low values of the size parameters (number of items, number of scenarios), quickly grow into large numerical problems where exact resolution is impossible. When this happens, we look at the improvements brought by the SDP relaxations over the LP relaxation.

References

[AG08] R. Lopez A. Gaivoronski, A. Lisser. Knapsack problem with probability constraints. Technical Report RR1498, LRI, July 2008.

[CB98] Amy Mainville Cohn and Cynthia Barnhart. The stochastic knapsack problem with random weights: A heuristic approach to robust transportation planning. In *Proceedings from TRISTAN III*, San Juan, Puerto Rico, 1998.

[DGV04] Brian C. Dean, Michel X. Goemans, and Jan Vondrak. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 208–217, Washington, DC, USA, 2004. IEEE Computer Society.

[GJ79] M. R. Garey and D. S. Johnson. *Computer and Intractability.* W. H. Freeman and Company, New York, 1979.

[GLS88] Martin Grötschel, Lászlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics.* Springer, 1988.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[HR98] C. Helmberg and F. Rendl. Solving quadratic (0,1)- problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82:291–315, 1998.

[IK75] Oscar H. Ibarra and Chul E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22(4):463–468, 1975.

[KP98] Anton J. Kleywegt and Jason D. Papastavrou. The dynamic and stochastic knapsack problem. *Oper. Res.*, 46(1):17–35, 1998.

[Pis07] David Pisinger. The quadratic knapsack problem-a survey. *Discrete Appl. Math.*, 155(5):623–648, 2007.

[SS06] David B. Shmoys and Chaitanya Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *J. ACM*, 53(6):978–1012, 2006.