



HAL
open science

Decision-making with uncertain data: Bayesian linear programming approach

Ayeley Tchangani

► **To cite this version:**

Ayeley Tchangani. Decision-making with uncertain data: Bayesian linear programming approach. Journal of Intelligent Manufacturing, 2004, 15 (1), pp.17-27. 10.1023/B:JIMS.0000010072.16604.04 . hal-02134734

HAL Id: hal-02134734

<https://hal.science/hal-02134734>

Submitted on 20 May 2019

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.




Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of some Toulouse researchers and makes it freely available over the web where possible.

This is an author's version published in: <https://oatao.univ-toulouse.fr/23641>

Official URL : <https://doi.org/10.1023/B:JIMS.0000010072.16604.04>

To cite this version :

Tchangani, Ayeley  *Decision-making with uncertain data: Bayesian linear programming approach.* (2004) *Journal of Intelligent Manufacturing*, 15 (1). 17-27. ISSN 0956-5515

Any correspondence concerning this service should be sent to the repository administrator:

tech-oatao@listes-diff.inp-toulouse.fr

Decision-making with uncertain data: Bayesian linear programming approach

A. P. TCHANGANI

EPA LGP, Dpt. GEII, IUT de Tarbes, Université Toulouse III 1, rue Lautréamont, 65016 Tarbes Cedex, France

EPA LGP, 47, Av. d'Azereix, B.P. 1629, 65016 Tarbes, France

E mail: tchangani@geii.iut tarbes.fr

This paper deals with decision making in a real time optimization context under uncertain data by linking Bayesian networks (BN) techniques (for uncertainties modeling) and linear programming (LP, for optimization scheme) into a single framework. It is supposed that some external events sensed in real time are susceptible to give relevant information about data. BN consists in graphical representation of probabilistic relationship between variables of a knowledge system and so permit to take into account uncertainty in an expert system by bringing together the classical artificial intelligence (AI) approach and statistics approach. They will be used to estimate numerical values of parameters subjected to the influence of random events for a linear programming program that perform optimization process in order to select optimal values of decision variables of a certain real time decision making system.

Keywords: Linear programming, Bayesian networks, decision making, decision support system, real time optimization

1. Introduction and statement of the problem

1.1. Decision-making by mathematical programming

Any organization, from a single person to a multi-national company through small companies, factories, etc., make always decisions (investment, resources acquisition, production planning, etc.). Decisions are made by choosing numerical values of some free variables called decision variables or control variables (production level for a given period, stock level, investment budget level, etc.) in order to optimize (minimize or maximize) some objectives (maximize profit, minimize cost, minimize risk, etc.) when respecting some constraints due to necessarily limitation of available resources (budget limitation, lack of qualified manpower, etc.) and some technical specifications. For instance, a production manager of a given company must establish a production plan by choosing

the level of each item to be produced for a given period (a day, a week or a month) using informations given by commercial and/or marketing department; a project manager must establish the planning of its crews respecting social and labor laws constraints; a logistic engineer must do the planning of reception dates of the resources to be used in a production process according to some requirements; and so on. These problems are well formulated when all the parameters are specified by the so-called mathematical programming (also known as optimization) problem in the literature. Basically, a mathematical programming problem consists of the following:

$$\begin{cases} \min / \max_x & f(x) \\ & G(x) \leq 0 \\ & x \in \mathcal{D} \subseteq \mathbb{R}^n \end{cases} \quad (1)$$

where $x \in \mathbb{R}^n$ is decision variables vector, $G(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is constraints vector, $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$

is the objective function to be optimized and \mathcal{D} is the admissible set of decision variables. Here \mathbb{R} and \mathbb{R}^d denote the set of real numbers and the real space vector of dimension d respectively. System (1) can represent a wide range of practical problem in engineering, social sciences, economics, management sciences, etc. According to the nature of functions f , G , the set \mathcal{D} , and decision variables x , the problem is known as linear programming, nonlinear programming, integer programming, mix integer programming, convex optimization, quadratic optimization (see Dantzig, 1963; Gill *et al.*, 1981; Hiriart-Urruty and Lemaréchal, 1993; Luenberger, 1984; Mangasarian, 1969; Moré and Wright, 1993; Nesterov and Nemiroski, 1994; Rockafellar, 1970; Teghem, 1996; Winston, 1994; Wright, 1996 and references therein for more details on theory, classification, algorithms, and applications).

In this paper we will consider the linear programming problem, that is

$$\left\{ \begin{array}{l} \min / \max_x c^T x, \\ Ax \leq b \\ x_{\min} \leq x \leq x_{\max} \end{array} \right., \quad x \in \mathcal{D} \subseteq \mathbb{R}^n \quad (2)$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ is often called technical matrix, $c \in \mathbb{R}^n$ is the objective vector, $b \in \mathbb{R}^m$ the right-hand side vector, $x_{\min}, x_{\max} \in \mathbb{R}^n$ the lower and the upper bound of decision variables, respectively. This problem is well known and widely studied in the literature in the theoretical view point as well as practical view point. Very efficient algorithms now exist to solve this problem ranging from classical simplex (Dantzig, 1963) algorithm to recently interior point algorithms (Nesterov and Nemiroski, 1994; Wright, 1996).

But the efficiency of a decision-making process based on former formulation is strongly related to the reliability of model parameters (A, b, c, x_{\min} and x_{\max}). In practice these parameters are subjected to uncertainties and are provided most of time by experts from different domains (production managers, specialists of marketing, logistic engineers, etc.). These specialists are not able, most of time, to give precise values of these parameters without relating them to some external random event E . For a production system where x consists in quantities of different items to be produced, the objective vector, c , may be

provided by commercial or marketing specialists according to their knowledge of the market in terms of: the prize of product P will be high (α money units), low (β money units), medium (γ money units), etc. if the event E is observed. This formulation is useless for the decision maker in terms of solving the problem (2) and so it is necessary to have a model that translate these uncertainties in numerical values. There are two main ways to take into account the problems of uncertainties: fuzzy logic theory approach that deals with imprecise (vagueness) variables or events and probability theory that is concerned with precise events but subjected to randomness. Next paragraph is devoted to this modeling problem.

1.2. Bayesian linear programming approach

The uncertainties we are dealing with in this paper are that due to randomness of some parameters as opposed to vagueness. Classical approach to dealing with randomness in linear programming problems is the so called stochastic programming approach (see Teghem, 1996, and references therein). This approach is essentially based on ‘‘here and now’’ paradigm, that is optimization problems must be solved with *a priori* probability distribution of random parameters without taking into account possible occurrence of some events that could give information on these parameters. Here we will consider the existence of possible events (or parameters) whose observation may give an idea of actual values of parameters of the problem. The mathematical tool used to deal with this causality relationship is Bayesian networks (BN). BN (Becker and Naïm, 1999; Jensen, 1999) are tools that can put together probabilistic causal relationships between parameters and events. This tool permits one to integrate uncertainty into an expert knowledge systems in terms of probability of occurrence of an event knowing that some particular event occurred. BN seems to be particularly interesting as they are able to derive *a priori* unknown relationship (Becker and Naïm, 1999) between variables; they can be used to estimate the parameters of system (2) subjected to the occurrence of some events. When this approach is used to estimate parameters of a linear programming problem, we will refer to as Bayesian linear programming. Bayesian linear programming model is well suited in the context of real time decision making where events that have significant effect on particular parameters of linear programming model

can be known with a good precision (presence of concurrents on the market of a product, reliability of production resource, weather of the day, etc.). Furthermore a decision systems support (DSS) based on Bayesian approach in real time production planning for instance can easily integrate different functional components of an enterprise such as reliability analysis (Tchangani, 2001) of production resources and marketing.

The remainder of the paper is organized as follows: in the second section a rapid presentation of BN is given including algorithms and existing software. The third section is devoted to the process of linking Bayesian network and linear programming into a single framework in order to solve a decision making problem under uncertain data. The last section shows the use of the developed method by considering two simple examples.

2. Rapid presentation of BN

2.1. Definition of BN

BN derive from convergence of statistical methods that permit one to go from information (data) to knowledge (probability laws, relationship between variables,...) and artificial intelligence (AI) that permit computers to deal with knowledge (not only information) (see, for example, Becker and Naïm, 1999). The terminology BN comes from the works by Bayes (1763/1958) in the eighteenth century. Its actual development is due to Pearl (1988). The main purpose of BN is to integrate uncertainty in expert system. Indeed, an expert, most of time, has only an approximative knowledge of the system that he or she formulates in terms like: *A* has an influence on *B*; if *B* is observed, there exists a great chance that *C* occurs; and so on. In the other hand, there are data (e.g., measurements) that contain some information which must be transformed into causality between variables. Thanks to probability, BN will solve this duality problem.

BN consists in a graphical representation of causality relationship between a cause and its effects. Figure 1 means that *A* is the cause and *B* its effect. But as this causality relationship is not strict, the next step is to quantify it by giving the probability of occurrence of *B* when *A* is realized. So a BN consists of an oriented graph where nodes represent variables



Fig. 1. Causality relationship in BN.

and oriented arcs represent the causality relationship and a set of some probabilities. The rigorous definition of a BN is given below:

Definition 1 (Jensen, 1999)

A Bayesian network consists of the following.

- *A set X of variables and a set of directed edges between variables.*
- *Each state has a finite set of mutual exclusive states.*
- *The variables together with the directed edge form a directed acyclic graph.*
- *To each variable X_i with parents¹ contained in the set $C(X_i)$, there is attached a conditional probability table $P(X_i/C(X_i))$.²*

Following assumptions (Becker and Naïm, 1999) are added for mathematical convenience.

- (1) Subjective probabilities (that used by an expert to describe relationship between variables) are considered as mathematical probabilities
- (2) Frequencies (table of measures for example) are considered as mathematical probabilities
- (3) The causality graph is representative of some probability law.

2.2. Algorithms and software related to BN

2.2.1. Algorithms

The main purpose of BN is to propagate a certain knowledge of the state of one or more particular nodes through the network so that one can get how the beliefs of the expert in the BN will change. So, given a BN it returns to compute the following conditional probability

$$P(X_i/Y) \quad \text{where } Y \subset X, X_i \notin Y$$

Using the properties of chains, trees, networks, d-separation concept (see Jensen, 1999), and the properties of conditional probabilities algorithms are derived that propagate a certain knowledge in the BN. For example for chain of length n , if X_i is downstream of X_j but not a direct relative, then

$$P(X_i/X_j) = \sum_{X_{i-1}} P(X_i/X_{i-1})P(X_{i-1}/X_j)$$

If X_{i-1} is a direct relative of X_j then stop if not decompose $P(X_{i-1}/X_j)$ as previously. For other forms (trees, general networks), it is more complicated but there exist algorithms based on other well-known algorithms of networks such as: maximum flow, short path, maximal weight trees, etc. For more information one can consult specialized literature as Becker and Naim (1999), Jensen (1999) and references therein.

2.2.2. Learning

BN constitutes a mode of knowledge representation founded on the description of relationship between variables in a given domain. If we consider knowledge as relationship between variables and information as the fact of a given situation, then the inference is the way to go from knowledge model to conclusion by computing some probabilities. The problem then is how to compute a knowledge model in order to use it in the future? This can be done in different ways: by expert knowledge that is known as knowledge acquisition, by doing experiences and measuring some data that will be used to fit some model, that is statistical or Bayesian approach. We will briefly recall in the following the two last cases. These two methods consider the best approximation in some sense of distribution of probabilities (frequencies of some events). The two fundamental components in BN are relationships between variables represented by arcs that we will refer to as the structure of the BN and the conditional probabilities referred to as parameters. Learning process is then to compute these two components. If the structure is known *a priori*, statistics and Bayesian analysis can be used to determine the parameters. In other hand the same techniques can be used to choose the best structure in some ‘‘sense’’ among a given set of structures. These two techniques will be reviewed in the following sections.

Notations

For a given BN with nodes $(X_i)_{1 \leq i \leq n}$, we have:

- the structure of the BN (acyclic directed graph),
- Θ , the parameters set of the BN,
- D , base of examples considered for learning,
- n , number of nodes,
- r_i , the number of states of X_i ,

- q_i , the number of states of the parents of X_i considered as a node,
- $N_{i,j,k}$, the number of examples in D for which X_i is in the state k knowing that its parents are in the state j ,
- $\theta_{i,j,k}$, the probability that the node X_i is in the state k knowing that its parents are in the state j

$$\theta_{i,j,k} = P(X_i = x_i^k / C(X_i) = S_i^j)$$

where $S_i^j \in \mathbf{S}_i$ (the set of all mutual exclusive state of $C(X_i)$).

Learning process is then to compute the parameters and/or the structure of the BN in order to optimize some measure of the distribution of the base of examples D . The main measure used in the literature is the likelihood or log-likelihood of the distribution and so learning process is to compute components that maximize this likelihood function.

Definition 2

The likelihood of a probability distribution P funded on N observations of X is given by

$$L(P, N) = \prod_{x \in X} P(x)^{N(x)}$$

where $N(x)$ is the frequency of x , and the log-likelihood is

$$L(P, N) = - \sum_{x \in X} N(x) \log(P(x))$$

Based on these functions some learning algorithms can be derived.

Statistical learning of parameters

According to previous notations, one can show that, the log-likelihood function is given by a function Φ_D of Θ :

$$\Phi_D(\Theta) = - \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{i,j,k} \log(\theta_{i,j,k})$$

and the problem of learning is to choose Θ in order to maximize $\Phi_D(\Theta)$.

Statistical learning of the structure

Here the problem consists of choosing the BN that give the best approximation of the base of examples. And the previous methodology can be used to this end. The problem is that without some constraints this can lead to a highly interconnected networks; so one

must pay attention and add some constraints in order to eliminate for example connections with insignificant probabilities.

Bayesian learning of parameters

Let consider a BN with fixed structure and parameters Θ and let $P(\Theta, X_1, X_2, \dots, X_n)$ be the joint probability law of parameters Θ and variables $(X_i)_{i=1,n}$ and let D be the associated base of examples; then the formula of Bayesian learning is

$$P(\Theta/D) = P(D/\Theta)P(\Theta)$$

At this stage different learning algorithms can be derived. For instance, the maximum *a posteriori* is to choose Θ in order to maximize the previous expression. If there is no *a priori* knowledge on Θ , that is, one considers $P(\Theta)$ to be uniform, this is equivalent to maximum likelihood approach. Other methods can be found in Becker and Naïm (1999) and references therein.

Bayesian learning of structure

The Bayesian learning of structure lies on the same principles as parameters learning (see Becker and Naïm, 1999, for more).

2.2.3. Existing BN software

The computation of parameters and the computation of prediction given that some events occur by hand can be heavy if the BN is large (many variables interconnected). To overcome this and because of great use of BN in many domain such as: data-mining, diagnostics, planning, banks, finance and defense (Becker and Naïm, 1999; Jensen, 1999) to name few, some softwares are developed to aid in quick modelling and analysis of BN. The leader in this domain is probably the company Hugin that develop a graphical oriented software Hugin Explorer. A free version Hugin Lite is available for download from the website of the company; Microsoft proposes MSBN, a graphical tool for constructing networks. For more information on the developer related to BN, one can consult Becker and Naïm (1999).

3. Bayesian linear programming

This section is devoted to the methodology of bringing together BN and linear programming in a single framework in order to develop a decision

support system with uncertain data. As presented in Section 1, the data of the decision process are matrix A , and vectors b, c, x_{\min}, x_{\max} . These data may depend randomly on environmental variables or events. The objective vector c may depend on some variables of the market; technical matrix A and resource limitation vector b as well as bounds x_{\min}, x_{\max} may depend on market and/or some technical variables such failure of machines, transportation's modes, availability of specialized crews and so on in the context of production planning for instance. In the following paragraph a model of these random parameters will be derived.

3.1. Modeling of parameters $A, b, c, x_{\min}, x_{\max}$

The parameters $A, b, c, x_{\min}, x_{\max}$ are random variables as effects of random environmental events. These events can be of different nature as they act on parameters of different nature. For instance, events that have an effect on the change of the prize (e.g., entrance of a new supplier) are not the same as events that may change the time of execution of a particular task (failure of a machine) in a factory. But for the sake of simplicity we will regroup these causes in the same set denoted by $\mathbf{E} = \{E_i\}_{i=1,N}$ where N is the total number of identified events. These events constitute the top level of the Bayesian network. To give a model of parameters, we consider that each entry of matrix A and each entry of vectors b, c, x_{\min}, x_{\max} is a random variable that can take a finite number of states and each state is associated with a numerical values, that is

$$\begin{aligned} a_{ij} &\in \left\{ a_{ij}^k \right\}_{k=1, n_{a_{ij}}}, & 1 \leq i \leq n; 1 \leq j \leq m \\ b_i &\in \left\{ b_i^k \right\}_{k=1, n_{b_i}}, & 1 \leq i \leq m \\ c_i &\in \left\{ c_i^k \right\}_{k=1, n_{c_i}}, & 1 \leq i \leq n \\ x_{\min,i} &\in \left\{ x_{\min,i}^k \right\}_{k=1, n_{x_{\min,i}}}, & 1 \leq i \leq n \\ x_{\max,i} &\in \left\{ x_{\max,i}^k \right\}_{k=1, n_{x_{\max,i}}}, & 1 \leq i \leq n \end{aligned}$$

where $n_{a_{ij}}, n_{b_i}, n_{c_i}, n_{x_{\min,i}}, n_{x_{\max,i}}$ are number of states for the corresponding coefficient respectively and one can have, for example, $n_{a_{ij}} \neq n_{a_{ip}}$ for $j \neq p$. We will suppose also that parameters do not have causal relationship between them (they depend only on top level events, \mathbf{E}) whereas the causes can have relationship between each other, that is $C(E_i) \subseteq \mathbf{E}$ (possibly empty), and

$C(\times) \subseteq \mathbf{E}$, where \times can represent any coefficient of parameters $A, b, c, x_{\min}, x_{\max}$.

Remark 1

Notice that some of the previous coefficients may not depend on external events, that is, they are known with certainty. From now, \times will generically represent any coefficients of parameters $A, b, c, x_{\min}, x_{\max}$.

The next section shows how one can bring BN and linear programming together through three principal units.

3.2. Different units in Bayesian linear programming

The link between BN and linear programming can be organized around three principal units: BN unit, parameters selection unit and the linear programming unit. The two first units are related to causal relationships and uncertainties modeling whereas the last one is a pure computation unit. A fourth unit called decision maker (DM) is added to analyze the obtained solution and decide if it is acceptable or some modifications in the uncertainties modelling process must be considered.

3.2.1. BN unit (BNU)

In BNU, one will construct causal relationships between external random events $E_i \in \mathbf{E}$ and parameters $A, b, c, x_{\min}, x_{\max}$ of the system. The main purpose of this unit is to propagate the information of occurrence of particular events $\{E_i\} \subseteq \mathbf{E}$ in the Bayesian network in order to modify the belief on parameters. It will then compute following conditional probabilities:

$$P(\times = \times^k / \mathbf{E}), \quad 1 \leq k \leq n_{\times}$$

This knowledge in terms of probabilities will be used to select in some manner appropriate parameters to be used in the optimization process. The following paragraph will consider some procedures to choose these parameters.

3.2.2. Parameters selection unit (PSU)

Once the previous conditional probabilities are computed, the parameters selection unit will use these informations to select particular value for coefficients of $A, b, c, x_{\min}, x_{\max}$ using some procedures. This unit may need an expert knowledge in order to derive a meaningful procedure of

parameters selection. Let us present some possible procedures for parameters selection.

Mean value procedure: This procedure is probably the first idea one may have for selecting parameters. Indeed, it considers the mean values of each components of parameters $A, b, c, x_{\min}, x_{\max}$ according to probabilities computed by the BNU, namely

$$\times = \sum_{k=1}^{n_{\times}} P(\times = \times^k / \mathbf{E}) \times^k$$

where \times is a generic parameter that can represent any coefficients of $A, b, c, x_{\min}, x_{\max}$.

Pessimistic/optimistic procedure: The mean value procedure is not necessarily the appropriate one. According to confidence one has in its experts, one can privilege a pessimistic or an optimistic behavior by selecting parameters that present minimum or maximum probability of occurrence, that is

$$\times = \arg \max_{\times^k} / \min P(\times = \times^k / \mathbf{E})$$

Remark 2

The two previous procedures can be combined. That is one procedure can be used for each group of parameters. Even in the case of second procedure, min approach can be used for a group of parameter whereas max procedure will be applied for another group. Of course, one can imagine other strategies for selecting parameters.

3.2.3. Linear programming unit (LPU)

This unit will use the previous selected coefficients to perform the optimization process. As mentioned in the introduction, linear programming is a well established subject nowadays and there are many solvers and modeling software devoted to it. For problems with less dimension one can use the function *linprog* of Matlab Optimization Toolbox. Other software combining modeling languages and a broad range of solvers are: GAMS, AMPL, MPL, MP XPress, etc.; or one can write its proper code. For more information on linear programming in general, see Moré and Wright (1993) or visit the NEOS website at <http://www-neos.mcs.anl.gov/neos/> where many informations can be found.

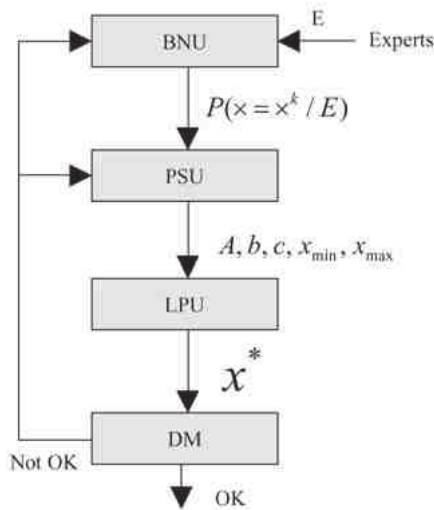


Fig. 2. Synopsis of decision making by Bayesian linear programming approach.

3.2.4. Summary

Figure 2 shows the synopsis of the method described previously. Experts of different domains concerned by the decision making process (for instance, commercial, maintenance, logistic) will supply the BNU by evidence of different events E ; BNU will propagate these evidence to compute different conditional probabilities, $P(x = x^k / E)$ and send them to PSU that will select the best value for x according to the procedure implemented. These values are supplied to LPU that will compute the optimal decision (or control) variables x^* and show it to decision maker (DM). DM will finally decide if this solution is acceptable or some adjustments are necessary in BNU and/or PSU.

4. Applications

In this section, the applicability of previous approach is considered through some small but instructive

applications that arise in the domain of marketing or production planning.

4.1. Example

This application is adapted from an example considered in page 727 of Winston (1994). It concerns a newspaper vendor that must determine each day how many papers to order. The vendor is sure to sell between 6 and 10 papers every day. This problem was formulated in terms of state-of-the-world decision-making problem in Winston (1994) and different criteria (Maximin Reward, Maximax Reward, Minimax Regret and Expected Value) was used to make decision considering that each possibility (number of papers demanded) is equally likely. Figure 3 gives the rewards matrix of this example.

This decision problem can be formulated as a binary value programming by considering decision variables to be $x_i = 1$ ($i = 6, \dots, 10$) if action i (number of papers ordered) is decided and 0 otherwise. One can see from the reward matrix that if the state of the world (that is the demand) is known then the action that maximizes the reward is determined. Actually, if the demand $j \in \{6, 7, 8, 9, 10\}$ is observed, the optimal action is to order $i = j$ papers. Now let us modify equally likely hypothesis of Winston (1994) and consider that the vendor has identified some events that have a great influence on the demand, namely the day of the week (Sunday excluded) and the weather. In terms of BN, we have Fig. 4 where the variable demand has two parents, Day that consists in six states (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday) and Weather that, we suppose, has three states (Sunshine, Cloudy, Raining). Demand is a variable with five states ($D_1 = 6, D_2 = 7, D_3 = 8, D_4 = 9, D_5 = 10$). Let suppose that from experience the vendor establishes that conditional probabilities for demand according to the day of the week and the weather are given by Fig. 5.³

Papers ordered	Papers demanded				
	6	7	8	9	10
6	30	30	30	30	30
7	10	35	35	35	35
8	-10	15	40	40	40
9	-30	-5	20	45	45
10	-50	-25	0	25	50

Fig. 3. Rewards matrix.

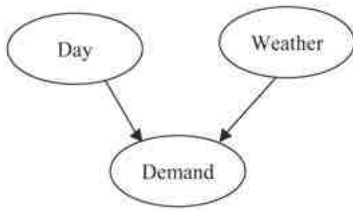


Fig. 4. BN of newspaper vendor problem.

This model is then introduced in a BN software Hugin Lite for simulation (Fig. 6). From evidence on the day and weather Hugin Lite will propagate this knowledge to determine probability of each state of demand. Then demand is selected by mean or max probability procedure. For instance, for a sunshine Saturday (see Fig. 6) Hugin Lite gives the following results:

Demand	$D_1(6)$	$D_2(7)$	$D_3(8)$	$D_4(9)$	$D_5(10)$
Probability	0.0	0.0	0.1	0.2	0.7

Maximum probability procedure will recommend that demand will be 10 and mean procedure gives 9.6 that is close to 10, so demand will be taken to be 10 and so is the order. From Monday to Friday and a sunshine weather we have the following results.

Demand	$D_1(6)$	$D_2(7)$	$D_3(8)$	$D_4(9)$	$D_5(10)$
Probability	0.1	0.2	0.4	0.2	0.1

Here the mean procedure and the max procedure lead to the same value of 8. Of course, other combination can be simulated.

4.2. Example 2

Here we consider a small production planning problem. A company is specialized in the production of two products P_1 and P_2 . A unit of P_1 is sold for 25 money units and a unit of P_2 is sold for 35 money units. A commercial study shows that demand X_1 and

Demand						
Weather			Sunshine			
Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
D1	0.1	0.1	0.1	0.1	0.1	0
D2	0.2	0.2	0.2	0.2	0.2	0
D3	0.4	0.4	0.4	0.4	0.4	0.1
D4	0.2	0.2	0.2	0.2	0.2	0.2
D5	0.1	0.1	0.1	0.1	0.1	0.7
Weather			Cloudy			
Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
D1	0.4	0.4	0.4	0.4	0.4	0.3
D2	0.3	0.3	0.3	0.3	0.3	0.3
D3	0.2	0.2	0.2	0.2	0.2	0.2
D4	0.1	0.1	0.1	0.1	0.1	0.1
D5	0	0	0	0	0	0.1
Weather			Raining			
Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
D1	0.6	0.6	0.6	0.6	0.6	0.5
D2	0.3	0.3	0.3	0.3	0.3	0.4
D3	0.1	0.1	0.1	0.1	0.1	0.1
D4	0	0	0	0	0	0
D5	0	0	0	0	0	0

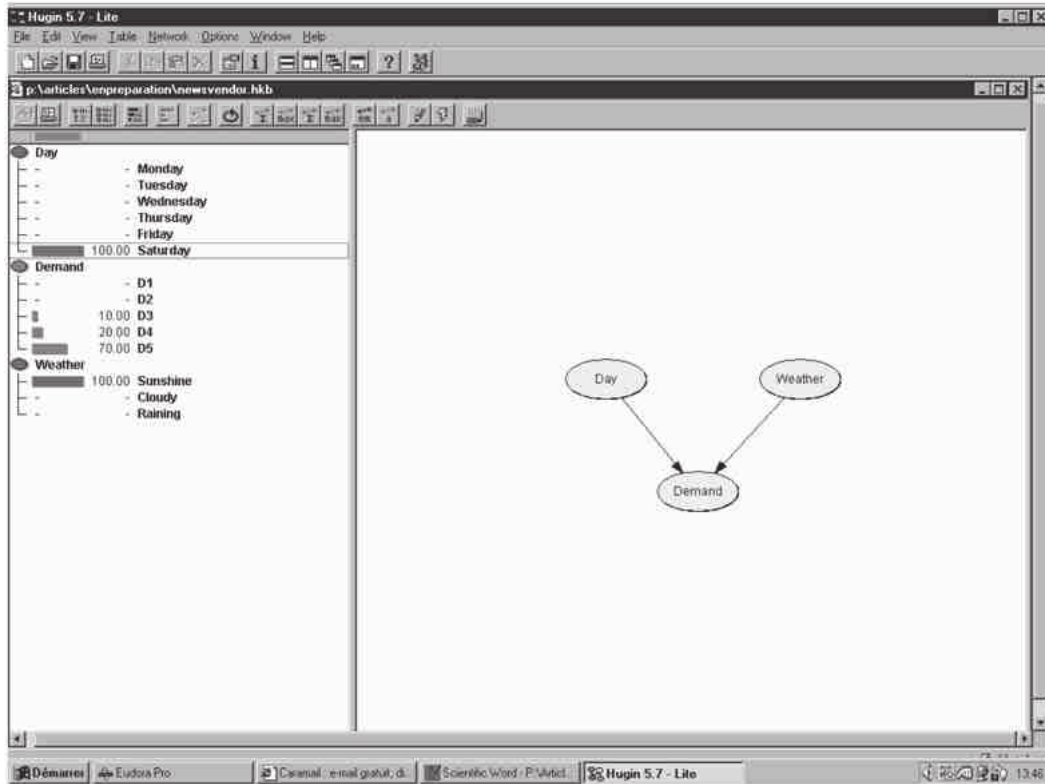


Fig. 6. News vendor BN in Hugin lite.

X_2 for these products can be considered as random discrete variables belonging to discrete set $\{Low_i, Nominal_i, High_i\}_{i=1,2}$ as a result of random events related to the market; it is considered that the most influential events are presence or not of competitors and the importance of customers. Low and High state of demands correspond to a variation of $\pm 20\%$ of the nominal demand. It is supposed that the competitors are represented by two states: Present/Not Present and customers by: Important/Not Important. Production of one unit of product P_1 requires t_1 machine time units and production of one unit of P_2 requires t_2 machine time units; t_1 and t_2 are random discrete variables belonging to the set $\{Large, Nominal\}$ as result of random events related to the state of the machine where Large corresponds to Nominal plus 10%. The machine can be in one of the states: Normal, Failure 1 and Failure 2. The period of planning is T time units. The purpose is to determine the quantity of each product to be produced during the

planning period T according to evidence of random events in order to maximize the profit.

4.2.1. Modeling

Linear programming modeling: Let us denote by x_1 and x_2 , respectively, the quantity of P_1 and P_2 to be produced. The linear problem is then

$$\begin{aligned} \max_{x_1, x_2} \quad & 25x_1 + 35x_2 \\ \text{s.t.} \quad & t_1x_1 + t_2x_2 \leq T \\ & 0 \leq x_1 \leq X_1 \\ & 0 \leq x_2 \leq X_2 \end{aligned}$$

In correspondence with system (2), we have

$$\begin{aligned} A &= [t_1 \ t_2], \quad b = T, \quad c^T = [25 \ 35], \\ x_{\min} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad x_{\max} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \end{aligned}$$

BN modeling: The BN of this system is given by Fig. 7 where X_i represent generically X_1 or X_2 and

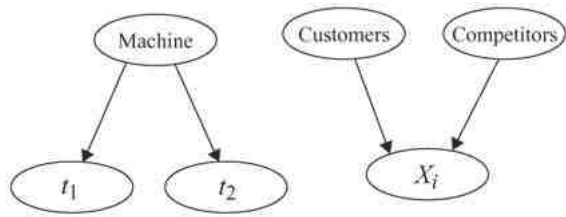


Fig. 7. Bayesian network of problem of Example 2.

conditional probabilities (considered just for simulation but consistent) for parameters t_1 , t_2 are given by Fig. 8 and those for X_i by Fig. 9.

4.2.2. Simulation

For simulation, we consider nominal values: $T = 160$ time units, nominal values for t_1 and t_2 , are respectively, 1 and 2 time units and nominal values of X_1 and X_2 , are respectively, 150 and 75. The solution without considering uncertainties is $x = [150 \mu 5]$ and the profit is 3925 money units. Figure 10 shows the output of linear programming unit for some configurations of random events; parameters are selected using mean procedure.

5. Conclusion and future works

The problem of decision making by linear programming approach under uncertain data is considered in this paper. An effort to link BN and linear programming in a single framework in order to solve this decision-making problem is presented. Because of availability of numerous software for modeling in the field of BN as well as in the field of linear programming, this approach can be useful for modeling and processing data for real time decision-making. Examples considered show the feasibility of this approach. The main drawback here is the separation of software dedicated to BN such as Hugin Lite and software dedicated to mathematical programming such as GAMS, AMPL, Matlab, XPress, etc. For instance in our second example, conditional probabilities are computed by Hugin Lite and optimization is done under Matlab with Optimization function linprog (of course, here one could solve optimization problem by hand!). An interface between software based on BN approach as Hugin Lite and optimization software or an integration of algorithms of BN and algorithms of optimization into a single software becomes necessary

t_1				
	Machine	Normal	Failure_1	Failure_2
Nominal		0.9	0.1	0.7
Large		0.1	0.9	0.3
t_2				
	Machine	Normal	Failure_1	Failure_2
Nominal		0.9	0.7	0.1
Large		0.1	0.3	0.9

Fig. 8. Conditional probabilities for parameters t_1 and t_2 according to the state of machine.

X_i					
	Competitors	Present		Not present	
	Customers	Important	Not important	Important	Not important
Low		0.2	0.5	0.2	0.3
Nominal		0.5	0.3	0.3	0.5
High		0.3	0.2	0.5	0.2

Fig. 9. Conditional probabilities for parameter X_i according to the state of competitors and customers.

	Machine	Normal	Normal	Normal	Failure_1	Failure_2
	Competitors	Present	Present	Not Present	Present	Present
	Customers	Important	Not important	Not important	Important	Not important
	Prob (Low)	0.2	0.5	0.2742	0.2	0.5
Probability of X_i	Prob (Nominal)	0.5	0.3	0.457	0.5	0.3
	Prob (High)	0.3	0.2	0.2688	0.3	0.2
	X_1	153	141	149.838	153	141
	X_2	76.5	70.5	74.919	76.5	70.5
Probability of t_1	Prob (Nominal)	0.9	0.9	0.9	0.1	0.7
	Prob (Large)	0.1	0.1	0.1	0.9	0.3
Probability of t_2	Prob (Nominal)	0.9	0.9	0.9	0.7	0.1
	Prob (Large)	0.1	0.1	0.1	0.3	0.9
	t_1	1.01	1.01	1.01	1.09	1.03
	t_2	2.02	2.02	2.02	2.06	2.18
Production	x_1	153	141	149.838	146.789	141
	x_2	2.7079	8.7079	4.2889	0	6.7752
	Profit	3919.8	3829.8	3669.7	3669.7	3762.1

Fig. 10. Bayesian linear programming results.

if one attempt to use this approach to solve real world problems. The learning process in the BN unit as well as model establishment in the parameters selection unit can get benefits with techniques as data mining, knowledge discovery in databases because many informations concerning parameters are often stored in databases. Future works will concentrate on these tasks.

Notes

1. Parents of a variable X_i are all variables that are causes of X_i .
2. Probability of X_i knowing $C(X_i)$.
3. Notice that these values do not come from a realistic experience but are chosen by the author with an effort to make them consistent and close to reality.

References

- Bayes, T. (1958) An essay towards solving a problem in the doctrine of chances. *Biometrika*, **46**, 293 298 (reprinted from an original paper of 1763).
- Becker, A. and Naïm P. (1999) *Les reseaux bayesiens*, Eyrolles, Paris.
- Dantzig, G. B. (1963) *Linear programming and extensions*, Princeton University Press, Princeton, NJ.

- Gill, P. E., Murray, W. and Wright, M. H. (1981) *Practical Optimization*, Academic Press.
- Hiriart Urruty, J. B. and Lemaréchal, C. (1993) *Convex Analysis and Minimization Algorithms*, Springer Verlag, Heidelberg.
- Jensen, F. V. (1999) *Lecture Notes on Bayesian Networks and Influence Diagrams*, Department of Computer Science, Aalborg University.
- Luenberger, D. G. (1984) *Linear and Nonlinear Programming*, 2nd edition, Addison Wesley.
- Mangasarian, O. L. (1969) *Nonlinear Programming*, McGraw Hill, New York.
- Moré, J. J. and Wright, S. J. (1993) *Optimization Software Guide*, Frontiers Appl. Math. 14, SIAM, Philadelphia.
- Nesterov, A. S. and Nemiroski, A. (1994) *Interior Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia.
- Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann.
- Rockafellar, R. T. (1970) *Convex Analysis*, Princeton University Press, Princeton, NJ.
- Tchangani, A. Ph. (2001) Reliability analysis using Bayesian networks. *Studies in Informatics and Control Journal*, **10**(3), 181 188.
- Teghem, J. (1996) *Programmation linéaire*, Editions, Ellipses/Editions de l'Université de Bruxelles.
- Winston, W. L. (1994) *Operations Research: Applications and Algorithms*, 3rd edition, Duxbury Press.
- Wright, S. J. (1996) *Primal Dual Interior Point Methods*, SIAM, Philadelphia.