Appl Intell (2007) 26:125–137
DOI 10.1007/s10489-006-0009-z

A decision support tool coupling a causal model and a multi-objective genetic algorithm

Ivan Blecic · Arnaldo Cecchini · Giuseppe A. Trunfio

Published online: 12 November 2006

Abstract A significant class of decision making problems consists of choosing actions, to be carried out simultaneously, in order to achieve a trade-off between different objectives. When such decisions concern complex systems, decision support tools including formal methods of reasoning and probabilistic models are of noteworthy helpfulness. These models are often built through learning procedures, based on an available knowledge base. Nevertheless, in many fields of application (e.g. when dealing with complex political, economic and social systems), it is frequently not possible to determine the model automatically, and this must then largely be derived from the opinions and value judgements expressed by domain experts. The BayMODE decision support tool (Bayesian Multi Objective Decision Environment), which we describe in this paper, operates precisely in such contexts. The principal component of the program is a multi-objective Decision Network, where actions are executed simultaneously. If the noisy-OR assumptions are applicable, such a the model has a reasonably small number of parameters, even when actions are represented as non-binary variables. This makes the model building procedure accessible and easy. Moreover, BayMODE operates with a multi-objective approach, which provides the decision maker with a set of non-dominated solutions, computed using a multi-objective genetic algorithm.

Keywords Bayesian networks · Decision networks · Influence diagrams · Multi-objective genetic algorithms

1 Introduction

Making decisions about complex systems can involve an analysis of interactions between actions and future events, which often proves to be a hard task for the human mind. Help in coping with such complexity can be supplied by normative Decision Support Systems (DSSs), implementing formal probabilistic methods of reasoning [1]. This kind of DSS allows the user to build a model, that is, an abstraction of the real world, in terms of interacting variables. These include at least *decision variables* (i.e. actions), which are under the decision maker's control, as well as *objective variables*. The latter are used to express the decision maker's preferences, often using utility functions [2]. Once the model is built, the DSS assists the decision maker in the identification of actions, by an appropriate formal method based on a decision theory and according to the specified preferences.

Since DSSs of this kind are increasingly requested in various domains of application, including medicine, economics, military, engineering and urban planning, they have been the subject of significant research efforts over the last decades. In particular, a vast amount of work has been carried out concerning the graphical probabilistic models related to the Bayesian Networks (BNs) [3], which are often adopted as a modeling technique and a reasoning engine for normative DSSs. These research efforts have been accompanied by the development of a number of BN software tools. At present, about fifty different programs and libraries are reported in

I. Blecic · A. Cecchini · G. A. Trunfio (✉)
Department of Architecture and Planning, University of Sassari, Palazzo del Pou Salit, Piazza Duomo, 6, I07041 Alghero (SS), Italy
e-mail: trunfio@uniss.it

I. Blecic
e-mail: ivan@uniss.it

A. Cecchini
e-mail: cecchini@uniss.it

[4, 5], and among these there are some popular tools such as Analytica [6], BayesiaLab, GeNIe and SMILE [7], Hugin [8], MSBNx, Netica. Mostly, the DSSs developed are for general purposes (e.g. [6–8]), while others have been conceived and optimized for specific fields of application (e.g. [9, 10]). The BayMODE program (Bayesian Multi Objective Decision Environment) described in this paper belongs to this latter category. It was developed in fact to operate in particular contexts where:

- the decision maker needs to simultaneously undertake different actions represented by numeric variables. An interesting specific case of this kind is when the decision maker has to allocate a quantity of resources for different actions. In this case, the variable associated with an action measures the 'investment' (in a wide sense e.g. economic, social, electoral, normative, consensus) the decision maker is willing to put into that specific action. For example, a town municipality should allocate the available budget for different simultaneous actions, which might be: the improvement of collective transport systems, the restoration of the historical center or the provision of facilities and services for citizens; at a certain point in a military conflict, the military planner might be in the position to have to decide how much effort to spend on an air raid and on a simultaneous ground attack; an entrepreneur might be in the situation of deciding how much of his profits to invest in the research and development of a new product and in exploring new markets for his current products.
- the decision maker needs to achieve a trade-off between different, often conflicting, objectives, which are probabilistically dependent on simultaneous actions. For example, the town municipality might wish for an increase in visiting tourists and the revitalization of the urban economy, as well as a decrease in traffic congestion and a reduction in the level of air pollution; the military planner might wish to win the conflict, but minimize the losses of his own military troops and in terms of civil population and civil infrastructures.
- the model must be built out of the knowledge provided by human experts. Indeed, in many fields of application, the cases collected are often insufficient or not suitable to determine the model with automatic learning procedures. In these situations the input information must largely be derived from experts' opinions and value judgements grounded on expert knowledge. For example, in the field of urban policy making, while it could be accepted that an increase in number of some particular categories of tourists can determine an increase in real-estate values, there are rarely data suitable to procedurally define the impact of the first event on the probability of the second. Likewise, it is difficult to obtain from data to what extent spending resources on the improvement of the collective transport system can impact the probability of a reduction in traffic congestion for a specific city. Other cases of models which cannot be automatically derived from the observation data are those involving events such as 'the development of a particular new product', 'the occurrence of a particular scientific discovery', 'the enactment of a specific piece of legislation' or 'the outbreak of a particular war'. For these kinds of event there is really no statistically significant history of occurrence, and the use of experts' knowledge in the phase of model construction appears mandatory.

BayMODE is based on the particular type of Decision Network (DN) [11] introduced in [12], coupled with a meta heuristic search algorithm. Such a special class of DN will be called a Simultaneous Decision Network (SDN), since there is no assumption of sequentiality of nodes representing actions.

Both the SDN used in the program and the coupled solution procedure are multi-objective. As argued in [13] and later in this paper, this has a number of advantages, especially when dealing with conflicting objectives, as often happens in the specific context of application outlined above. Thus, BayMODE's main outcome is a set of non-dominated solutions [14], each representing a *strategy* (i.e. a vector containing a numeric value for each action).

In real applications, incorporating expert knowledge into a standard DN can be arduous, since the number of conditional probabilities to be estimated for a single node grows exponentially with the number of its direct parents. When actions are not binary, the problem becomes even more relevant. For these reasons, BayMODE makes use of the well known noisy-OR canonical model of interaction [3, 15, 16], which provides various advantages deriving from the causal interpretation of the interaction between the model entities and the logarithmic reduction of the number of parameters. As will be shown later in the paper, in the field of application of BayMODE, such a model of interaction is often well justified. In the context of the noisy-OR assumption, the program offers a simple and convenient way to incorporate the effects of the simultaneous non-binary actions into the model. This is provided through the use of functional relationships, which link the decision variables to the probability of their effects. Such interaction patterns are then estimated in a semi-qualitative form by the experts with the aid of a specifically conceived user interface.

With respect to existing general purpose Bayesian DSSs, BayMODE operates more effectively in the context for which it was specifically designed. The reasons for this lie in different concurrent characteristics, namely: (i) it allows modeling of the probabilistic effect of many numeric actions applied in parallel; (ii) the approximate effects of such non-binary actions can be estimated by domain experts and then included in the model in a simple and effective way; (iii) it offers an intrinsic multi-objective perspective, which results in providing the decision maker with a set of non-dominated solutions with a single run of the search algorithm.

The outline of the paper is as follows. In Section 2 we present BayMODE's underlying model. In Section 3 we discuss the problem of searching for a trade-off strategy with the use of a multi-objective genetic algorithm. The section assumes some general knowledge of genetic algorithms. Subsequently, we present an example application of a policy-making case study in Section 4. The paper ends with Section 5, presenting some conclusive considerations.

2 The causal model

BayMODE allows the construction of a causal model whose entities are represented by variables which we will indicate by uppercase letters (e.g. Y) or indexed uppercase letters (e.g. X_i). A specific value of a variable will be denoted by lowercase letters (i.e. x). The set of all possible values of a variable Y will be indicated as $D(Y)$. A set of variables will be indicated by calligraphic style letters (e.g. $\mathcal{Y}$). The general scheme of this model is shown in Fig. 1 and it includes:

- a set $\mathcal{A}$ of m actions which are modeled as real number variables. The value a of an action $A \in \mathcal{A}$ represents some physical or abstract quantity relevant to the modeled context (e.g. the effort spent in terms of energy, money, etc.). In general, the domain of definition of an action A is the discrete set $D(A) \subseteq [0, +\infty[$ that is typically a result of the discretization of a continuous interval. As will be shown later (see Section 2.2), under some conditions the model can easily and effectively be built even in the case of high cardinality of $D(A)$ for each action A;
- a set $\mathcal{V}$ of events which are modeled as binary variables. The two values of an event $Y \in \mathcal{V}$ will be denoted by the corresponding lower case letter y, with the meaning of 'occurrence', and negate lowercase letter $\neg y$. Hence, the domain of definition of an event Y is $D(Y) = \{y, \neg y\}$. In order to facilitate the process of elicitation of the model by the domain experts, the event set $\mathcal{V}$ is partitioned in the set $\mathcal{U}$ containing *exogenous events*, which cannot be influenced by any of the entities in the model, and the set $\mathcal{E}$ containing *endogenous events*, which can be influenced by all other events in $\mathcal{V}$ and by actions in $\mathcal{A}$;
- a set $\mathcal{C}$ of k objectives. Each $C \in \mathcal{C}$ is a variable defined in the set of real numbers $\Re$, reflecting the value of an objective function g_C.

An element of the set $\prod_{Y \in \mathcal{Y}} D(Y)$ will be defined as a *configuration* of a set of variables $\mathcal{Y}$. In particular, a configuration α of the set of actions $\mathcal{A}$ will be called *strategy*.

The model of interaction used is a BN, integrated with decision nodes and objective nodes, that is, a Decision Network (DN). It is worth noting that, in contrast with the more frequent utilization of DNs, in this case the multiple decisions must be made in parallel and not sequentially. In other words, in the DNs used in this paper there is no precedence among decision nodes, and they are not dependent on any other model entity, so the model can be called Simultaneous Decision Network (SDN) (or Simultaneous Influence Diagram as in [17]).

As in every BN, the causal model can be represented by a directed acyclic graph consisting of a set of nodes and the links between these nodes (see Fig. 1). Each node is associated with a variable Y representing a model entity. For simplicity, we will refer interchangeably to nodes and their associated variables. Each variable Y has a parent set $\pi(Y)$ in the network. Actions and exogenous events have no ingoing arcs, i.e., their parent sets are always empty. Endogenous events can have ingoing arcs representing probabilistic dependence, whereas objectives can have ingoing arcs representing functional dependence. In the DN, nodes representing objectives do not have children.

In particular, for each $E \in \mathcal{E}$, the distribution $P(E \mid \pi(E))$, where $\pi(E) \subseteq \mathcal{V} \cup \mathcal{A} \setminus \{E\}$, specifies the probability of each value of E, given every possible assigned value of $\pi(E)$. If the domains of definition of E and $\pi(E)$ are finite, such a distribution is called a Conditional Probability Table (CPT).

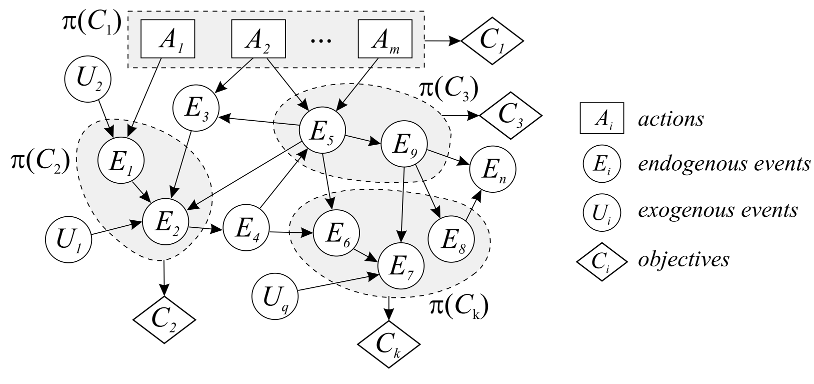

Fig. 1 A multi-objective SDN scheme and the entities involved

Each objective C assumes a value given by an objective function $g_C: D(\pi(C)) \rightarrow \Re$, where $D(\pi(C)) = \Pi_{X \in \pi(C)} D(X)$. By definition, as shown in Fig. 1, the parent set of the objective C_1 is the set of actions, i.e., $\pi(C_1) = \mathcal{A}$, while for each objective C_i, $i = 2, \ldots, k$, a specific parent set $\pi(C_i) \subseteq \mathcal{E}$ is defined as part of the model.

In order to simplify formalization, an action variable A_i can be viewed, with respect to a strategy α, as a random variable whose probability $P_\alpha(A_i)$ is defined as:

$$P_\alpha(A_i) = \begin{cases} 1, & \text{if } A_i = a_i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Thus, temporarily neglecting the objective variables, the SDN can be seen as a BN representing the following joint probability:

$$P_\alpha(\mathcal{V}, \mathcal{A}) = \prod_{Y \in \mathcal{V}} P(Y \mid \pi(Y)) \prod_{i=1}^{m} P_\alpha(A_i) \tag{2}$$

The *expected value* of the objective variable $C \in \mathcal{C}$, in the case strategy α gets activated, is:

$$E_\alpha(C) = \sum_{\pi(C)} P_\alpha(\pi(C))\, g_C(\pi(C)) \tag{3}$$

where $P_\alpha(\pi(C))$ denotes the marginal probability of the variables $\pi(C)$ in the BN, that is:

$$P_\alpha(\pi(C)) = \sum_{\mathcal{A} \cup \mathcal{V} \setminus \pi(C)} P_\alpha(\mathcal{V}, \mathcal{A}) \tag{4}$$

with summation extended to all possible values in $\mathcal{A} \cup \mathcal{V} \setminus \pi(C)$.

The BayMODE program also allows the definition of scenarios, where a *scenario* s is defined as a configuration of a subset $\mathcal{S}$ of $\mathcal{V}$. The scenarios thus defined are of particular use in the evaluation of the performance of a specific strategy α, assuming that evidence is given in terms of occurrence or non-occurrence of a subset of events in $\mathcal{V}$. One of the possible situations would be that of a decision maker seeking to know what the best actions to undertake are, under the assumption of occurrence of a specific subset of exogenous events. Here, for each scenario s, if the strategy α has been applied, the expected value of the objective variable $C \in \mathcal{C}$ is:

$$E_{\alpha,s}(C) = \sum_{\pi(C)} P_\alpha(\pi(C) \mid s)\, g_C(\pi(C)) \tag{5}$$

where $P_\alpha(\pi(C) \mid s)$ is the conditional probability of $\pi(C)$ given s.

2.1 Multi-attribute utility versus multi-objective approach

In contrast with more frequent DNs with a single terminal objective node, in the model adopted in BayMODE the multiple objective nodes are retained. This is in accordance with what has been suggested in [13], where it is argued that acting in a multi-objective perspective with DNs can provide great advantages for the whole decision process.

When multiple objectives C_i are combined in a single terminal value node, this is normally accomplished through a multi-attribute utility function [2, 18]. In practice, a multi-objective problem is transformed into a single-objective problem of expected utility maximization. Frequently, under certain conditions that have been axiomatized [18], the multi-attribute utility function is constructed as a product or sum of the objective variables, with some parameters (i.e. weights) grounded on the decision maker's preferences.

The critical point in this approach is the definition of a suitable multi-attribute utility function [19, 20], which may incorporate many conflicting objectives, possibly with different physical meanings. In particular, there is usually high subjectivity in the choice of both the form and parameters of the multi-attribute utility function. Moreover, assessing the utility function requires an explicit statement of preferences by the decision maker prior to the solution process: if the preferences change, the entire solution process must be repeated. Another issue is related to possible inaccuracy in the expected value of an objective: using a multi-attribute utility function, the entire output of the decision making process will be affected by that inaccuracy. Finally, while the solution process can potentially give valuable information about the system represented by the DN, this is precluded by combining all objectives in a single value.

The different approach adopted in the DN represented in Fig. 1, consists of maintaining the objectives multiple and emphasizing the generation of a range of solutions to be presented to the decision maker for consideration. The main purpose of this approach is to produce a set of *non-dominated* (or *efficient*) solutions [14] in the way explained later in Section 3. Then, the solution eventually chosen by the decision maker is obtained by examining and exploring the various trade-offs between objectives for the set of non-dominated solutions. This latter phase can be carried out in a systematized way as in the Surrogate Worth Tradeoff method [21].

In practice, the decision maker's preference is used only after the most computationally expensive phase of the entire decision support process, and avoiding the usual assumptions of utility theory or the specification of a multi-attribute utility function [21]. On the other hand, however, it must be observed that the choice of the final solution becomes indeed more difficult as the number of objectives and the number of non-dominated solutions grow.

2.2 Probabilistic dependence of events

BayMODE was specifically designed to be used by a panel of domain experts able to build a model from scratch, since the knowledge-base for automated or semi-automated model construction is unavailable. This suggested the adoption of a scheme of interactions between entities, which, in most cases, provides reliable and efficient incorporation of experts' knowledge.

Although the program allows the user to specify the complete CPT for each node in the model graph, this might be a hard task in some cases, since the number of CPT entries for a node X is exponential to the number of configurations of its parent set $\pi(X)$. The problem is particularly evident when one or more non-binary actions are in $\pi(X)$. In fact, the set $D(A)$ may contain few values (e.g. labeled values as in $D(A) = \{\textit{very low}, \textit{low}, \textit{medium}, \textit{high}, \textit{very high}\}$) and may also be defined by a very fine subdivision of a given continuous interval.

For the above reasons, BayMODE implements the most frequently accepted and widely applied solution to this problem, that is the noisy-OR gate [3,15,16], which has the main advantages of providing a logarithmic reduction in the number of model parameters, as well as a causal interpretation of the interactions. Moreover, in this context the program allows a simple and effective way to estimate and take into account the effects of non-binary actions.

The use of the noisy-OR model in BayMODE is justified, as in its field of application the hypothesis of independence of causes is often quite well verified. Coming back to the example of a town municipality formulated in Section 1, a domain expert may suggest that both investing resources in the renovation of the historical center and the currency exchange rate, can have an impact on the probability of the event 'increase in visiting tourists'. But it is likely that the two causal mechanisms act quite independently.

Moreover, in the case of models built completely by domain experts, using the noisy-OR gate can lead to appreciable outcomes even when the causes do not act perfectly independently. Indeed, as recently argued in [12], some empirical studies concerning models built by human experts show that even when distribution does not rigorously respect the noisy-OR hypothesis, the elicitation error affecting the parameters might be smaller than the elicitation error of the complete CPTs. This decrement of estimation errors could possibly be due to the significant reduction in the number of parameters combined with the causal interpretation of the interactions, which make the estimation task more suited for a human mind. As a conclusion, it seems reasonable to assume that in this field of application it is better to use a model

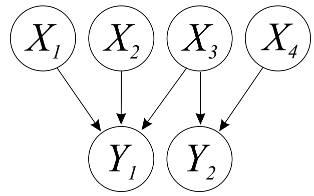

Fig. 2 Causes (X_i's) and effects (Y_i's)

that is to some extent imperfect (i.e. neglecting some small interactions), but with small elicitation errors, rather than a more complete but far too complex model, the parameters of which are in all probability affected by larger uncertainties.

It is also worth observing that the causal interpretation of the interactions provided by the noisy-OR gate, helps the domain experts to better identify the building blocks of the model [22]. This, in our experience, results in great advantages in terms of the overall quality of the decision process.

2.2.1 Interactions between events

With the purpose of briefly recalling the noisy-OR assumptions used in BayMODE as a favorite way to model the probabilistic dependence of events, let us say that the event represented by the variable X_i may be a cause of Y when it is *active* with respect to Y or, in other words, when it assumes a particular value $x_i^{\uparrow Y}$. For example, with reference to Fig. 2, if the binary variable X_3 represents an event, it may be that its occurrence is a cause of the effect Y_1 whereas non-occurrence is a cause of Y_2. It is worth noting that since in a large BN an event X_i may represent a cause for more than one effect, the activation value of X_i may be different for each effect. In general, given a configuration of $\pi(Y)$, we can define the set:

$$\pi^*(Y) = \{X \in \pi(Y) : X = x^{\uparrow Y}\} \tag{6}$$

to which all the X_i belong which are in the active state for Y. Formally, the basic noisy-OR assumptions are:

- each of the causes X_i is characterized by a parameter $p_i \in [0, 1]$ called *causal strength* with respect to its potential effect Y;
- the ability of each cause to be sufficient is independent from the presence of other causes (i.e. when effect Y has not been produced, each cause X_i has failed independently).

The parameter p_i is defined as the probability the effect Y is true, cause X_i being active and all other causes X_j, with $j \neq i$, inactive:

$$p_i = P\big(y \mid x_i^{\uparrow Y} \wedge \neg x_j^{\uparrow Y}\ \forall\, j \in [1, n],\, j \neq i\big) \tag{7}$$

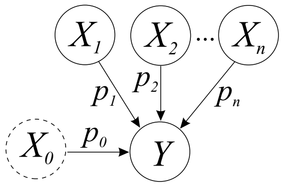

Fig. 3 The leaky noisy-OR model

The above hypothesis allows us to specify the entire CPT, given the non-empty activation set $\pi^*(Y)$, and only n parameters, in other words, to specify only a parameter p_i for every X_i. In particular, the probability of y is given by the following formula:

$$P(y \mid \pi^*(Y)) = 1 - \prod_{i:X_i \in \pi^*(Y)} (1 - p_i) \tag{8}$$

which is sufficient to derive the CPT of Y on its predecessors $X_1, X_2, \ldots, X_n$.

Since in practice it is almost impossible to enlist all possible causes which can produce effect Y, BayMODE uses the well-known extension of the binary Noisy-OR gate, called *leaky Noisy-OR* gate [15], which is suitable for situations where the effect can be produced even if all its explicit causes are inactive. This can be conceptualized by introducing an additional cause X_0 which is assumed to always be active (see Fig. 3). This cause is associated with an additional parameter $p_0 \in [0, 1]$, called the *leak probability*. It represents the probability that the effect Y is produced by the non-modeled causes, when all the modeled causes are inactive. Hence:

$$p_0 = P\big(y \mid \neg x_j^{\uparrow Y}\ \forall j \in [1, n]\big) \tag{9}$$

Since X_0 is always active, the CPT is expressed by the equation:

$$P(y \mid \pi^*(Y)) = 1 - (1 - p_0) \prod_{i:X_i \in \pi^*(Y)} (1 - p_i) \tag{10}$$

where, in this case, p_i is the probability that effect Y is produced by X_i, being inactive all the other modeled and non-modeled causes, while $\pi^*(Y)$ has the same meaning as in Eq. (8).

2.2.2 *Impact of actions on events*

Equation (10) refers to the situation where an event Y has only other events X_i as its explicit causes in the model. According to what was illustrated before, concerning the advantages of facilitating the elicitation phase, this can be extended to the situation represented in Fig. 4, where among the causes of Y there is also the action A. To this end, let us introduce an

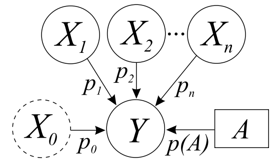

Fig. 4 Action causal strength

additional factor $1 - p(A)$ in Eq. (10), which becomes:

$$P(y \mid \pi^*(Y), A) = 1 - (1 - p_0)(1 - p(A)) \prod_{i:X_i \in \pi^*(Y)} (1 - p_i) \tag{11}$$

where $p(A)$ is the causal strength of the action A, with an obvious meaning in the context of the noisy-OR assumptions. During the elicitation process $p(A)$ can be given in a tabular form, that is, a strength value for each of the discrete values of A. A different way to provide $p(A)$ is through a function ψ representing an *interaction pattern*:

$$p(A) = \psi(A; q_1, \ldots, q_n) \tag{12}$$

where both ψ and the parameters $q_1, \ldots, q_n$ must be chosen during the modeling phase. In Fig. 5, few examples of strength functions included in BayMODE are represented. For instance, the function I allows the causal strength to be expressed in terms of the three parameters p_{inf}, a_α and a_β, that is, the asymptotic value of ψ, and values of the action corresponding to the strengths $\alpha\ p_{inf}$ and $\beta\ p_{inf}$, respectively. Returning to the example of the town municipality, a domain expert may estimate that spending resources on the renovation of the historical center acts on the event 'increase in visiting tourists' with a causal strength which follows the

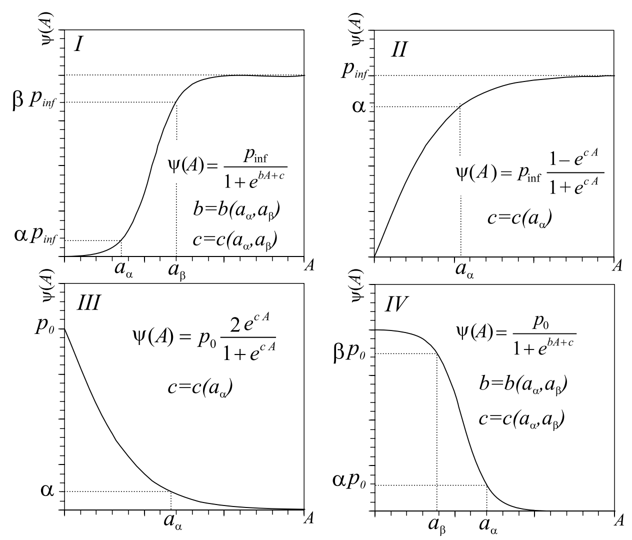

Fig. 5 Examples of causal strength functions depending on two (II and III) or three (I and IV) parameters

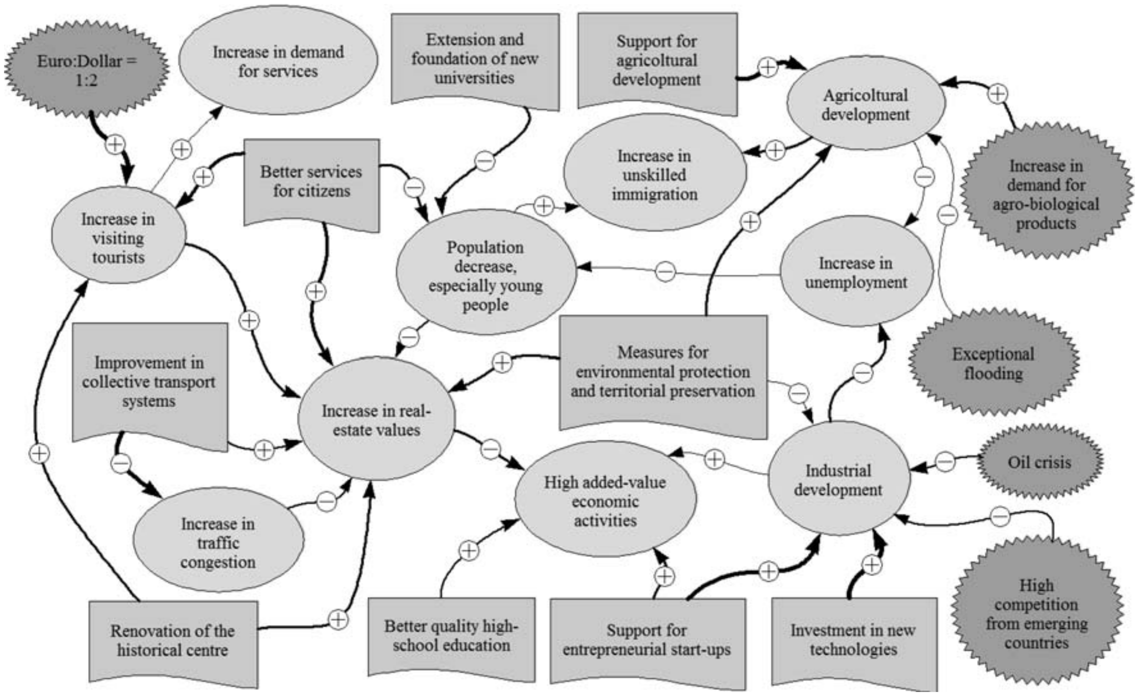

Fig. 6 The presented example graph from BayMODE

logistic curve I. Then, the expert should estimate the maximum causal strength p_{inf} inducted by that action, as well as the amount of resources corresponding, for example, to the 10% and the 90% of the causal strength p_{inf}.

Like many other Bayesian DSSs [4, 5], BayMODE was designed in a way to help experts be involved in a collaborative interactive phase of modeling, considerably assisted by the visual definition of the model in the form of a directed graph (see Fig. 6). In particular, for arcs linking actions and events the user can provide the causal strength both in tabular form and in terms of choice of the shape of the function with the associated parameters. During this phase, BayMODE interactively visualizes the chosen shape of the function, given its actual parameters.

3 Searching for best strategies

Once the model has been built, the subsequent step in the BayMODE-supported decision process is the computation of a set of strategies to be presented to the decision maker for consideration. In the line of principle, an optimal *strategy* $\alpha = (a_1, \ldots, a_m) \in \Lambda = \prod_{i=1}^{m} D(A_i)$, could be found with an exhaustive search, consisting of the computation of the expected values of the k objectives C_i in correspondence with all possible strategies.

With the aim of avoiding an inefficient exhaustive search, a number of methods have been conceived to find the optimal *policy* for DNs. Most algorithms concern standard DNs [11], which satisfy the *regularity* constraint (there must be a directed path containing all decision nodes), and the *no-forgetting* constraint (a decision node and its parents are parents to all subsequent decision nodes, that is, information available now should also be available later).

Roughly speaking, the existing methods can be classified as those operating directly on the given network, and those that perform the evaluation after transforming the network into a different representation. The well-known method described by Shachter [23] belongs to the first class, which consists of transforming the DN by successively removing nodes from the graph, like in stochastic dynamic programming [24], until the final objective node holds the value corresponding to the optimal policy. To the second class belong the methods which transform the diagram into a BN. This approach was first proposed by Cooper [25] and later by Shachter and Peot [26] and Zhang [27]. A related method transforms the DN into a junction tree that is suitable for probabilistic inference using the clique-tree propagation algorithm [28, 29].

Since all the algorithms for standard DNs adopt the regularity and no-forgetting constraints, they cannot be used for the SDN implemented in BayMODE. Similar considerations are valid for methods concerning non-standard DNs, the correctness of which relies on some specific properties of those DNs [30–32], which do not hold for SDNs.

On the other hand, in our case the exhaustive approach might turn out to be computationally infeasible. In fact, actions are represented by non-binary variables, and this potentially leads to a very large dimension of the decision space Λ.

For example, supposing that the m actions A_i belong to finite sets of definition $D(A_i)$ the cardinality of which is q, then the number of different strategies is $|\Lambda| = q^m$ (e.g. almost ten million strategies in the case of $q = 5$ values and $m = 10$ actions), while in the case of binary actions it equals $|\Lambda| = 2^m$ (e.g. about one thousand strategies for $m = 10$ actions).

To cope with such a high dimension of the search space, the approach followed in BayMODE consists of viewing the problem as a combinatorial optimization, for which an approximate solution is provided by a multi-objective meta heuristics. The use of heuristics is not new for BN-related problems [33–38], although, to our knowledge, multi-objective heuristics has never been used before. The problem can be conveniently formulated as a multi-objective search as follows:

$$\max_{\alpha \in \Lambda} E_{\alpha,s}(C_i), \quad i = 1 \ldots k \tag{13}$$

where $E_{\alpha,s}(C_i)$ is defined by Eq. (5) in correspondence with a given scenario s.

A more complete formulation may include some constraints. For example, if actions represent homogeneous quantities, it may prove convenient to impose constraints on the total amount such as:

$$\sum_{i=1}^{m} A_i \leq a_{max} \tag{14}$$

where $a_{\max}$ is the maximum total value allowed to be allocated for different available actions.

In the current version of BayMODE, the meta heuristics adopted for solving the (13) is a multi-objective Genetic Algorithm (GA) [39,40]. The GA is used to evolve a randomly initialized population, whose generic *chromosome* is an m-dimensional vector $\mathbf{s}$ representing an element $\alpha \in \Lambda$. The i-th element—or *gene*—of the chromosome is obtained as the binary encoding of A_i, using a suitable number of bits (optionally different for each action) and its interval of definition $D(A_i)$. Each chromosome can be decoded in a strategy α and, performing a BN inference, the objective functions can be computed.

In the multi-objective GA, to avoid the aggregation of multiple objectives in accordance with what was illustrated in Section 2.1, the comparison of two candidate solutions, with respect to different objectives, is achieved through Pareto's optimality and dominance concepts. In particular, considering the optimization problem (13), we say that a solution α (strongly) *dominates* the solution β if:

$$\forall\, i \;\; E_\alpha(C_i) \geq E_\beta(C_i) \wedge \exists\, j \;\; : E_\alpha(C_j) > E_\beta(C_j) \tag{15}$$

In other words, α dominates β if α is better or equivalent to β with respect to all objectives, and better in at least one objective. A non-dominated solution is optimal in the Pareto sense (i.e. no criterion can be improved without worsening at least one other criterion). Rather than a single solution, a search based on such a definition of optimum produces a set of non-dominated solutions, from which the decision maker should pick one.

The multi-objective GA adopted is the well-known NSGA-II [41], which has been extensively investigated and successfully tested [41–44]. The NSGA-II algorithm is based on the idea of transforming the objectives into a single fitness measure by the creation of a number of fronts, sorted according to non-domination. The fronts are created using Goldberg's 'non-dominated sorting' procedure [45], which works as follows:

1. all non-dominated individuals in the current population are inserted in the first front, which corresponds to the highest fitness;
2. these individuals are virtually removed from the population and the next set of non-dominated individuals are inserted in a second front, corresponding to the second-highest fitness;
3. phases 1–2 are reiterated until all of the individuals have been assigned a fitness rank.

When each front has been created, the so-called *crowding distances* (i.e., normalized distance to closest neighbors in the front in objective space) are assigned to its members, to be used in the next phase with the purpose of promoting a uniform sampling of the Pareto set.

Selection is performed by binary tournaments [41]: between two individuals the one with the lowest front number wins. If the individuals come from the same front, the one with the highest crowding distance wins, since a high distance to the closest neighbors indicates that the individual is located in a sparsely populated part of the front. If N is the size of the population, in each generation N new individuals are generated by a standard crossover with a predefined probability p_c. Then, a mutation with a probability p_m is applied to each offspring at each position in the chromosome. The algorithm is elitist, in the sense that out of the $2N$ individuals, the best N individuals are kept for the next generation. The constraints, like that in Eq. (14), are handled with the concept of *constrained dominance* proposed in [41]. It consists of a redefinition of the criterion (15), in such a way that any feasible solution belongs to a better non-domination front than any infeasible solution.

Execution of the NSGA-II algorithm requires the evaluation of the objectives corresponding to every individual in the population. This involves the use of a suitable inference procedure for the BN representing joint probability (2), in order to carry out the computation of $P_\alpha(\pi(C) \mid s)$.

BayMODE performs the evaluation of the objectives using the so-called *likelihood weighting* method [46, 47], which belongs to the class of approximate inference algorithms based on stochastic simulation. The method is a variation of the so-called probabilistic *logic sampling* [48], where: (*i*) repeated simulations of the world described by the BN are performed following the causal links, (*ii*) samples that are inconsistent with the evidence values are thrown away, and (*iii*) the probabilities of query nodes are estimated by counting the frequency with which relevant events occur in the sample. Unfortunately, if there is unlikely evidence, most generated samples will be inconsistent with that evidence and therefore wasted. In the *likelihood weighting* method, this problem of logic sampling is mitigated: instead of sampling evidence nodes, the observed value of the evidence variables is weighted by the likelihood of evidence.

The likelihood weighting algorithm was implemented in BayMODE for various reasons. First, the problem of unlikely evidence is not severe since the evidence (i.e. the values of random variables representing actions and the current scenario) is mainly concentrated in the roots of the BN graph (scenarios are mostly composed of variables in the set $\mathcal{U}$ of exogenous variables). Second, the method permits the control of accuracy, which increases with the number of samples generated and which is not affected by network topology and size. Indeed, with certain limitations, the GA does not require the exact evaluation of objectives in order to converge. This allows fast execution of the GA, even with large networks, because of the relatively small number of samples required for the evaluation of each individual.

4 An example application

The example application shown in this section refers to a policy-making exercise. The model graph is represented in Fig. 6, including actions (i.e. rectangles) and events (i.e. ovals). Table 1 reports all events included in the model and their estimated leak probabilities, as well as the estimated probabilities $\bar{p}$ assuming that no actions were undertaken. Table 2 reports the actions that in this model represent the efforts associated with their execution, defined in the interval [0, 10] and expressed in a homogeneous virtual unit. In Table 3 the characteristics of the causal strengths corresponding to the arcs in Fig. 6 are presented.

Interactions between events were modeled by the noisy-OR assumption as explained in Section 2.2. In the graph, a minus symbol labeling an arc from an event X to an event Y means $x^{\uparrow Y} = \neg x$, whereas a plus symbol means $x^{\uparrow Y} = x$ (see Section 2.2). A minus symbol labeling an arc from an action A to an event Y means that a decreasing causal strength function was used, whereas a plus symbol means that an increasing one was used (see also Fig. 5). The thickness of

Table 1 The set of events in the example presented (U_x denotes exogenous events, E_x denotes endogenous events, p_0 is the leak probability while $\bar{p}$ is the estimated probability corresponding to the null strategy)

Id	Description	p_0	$\bar{p}$
U_1	Increase in demand for agro-biological products	0.1	
U_2	Exceptional flooding	0.02	
U_3	High competition from emerging countries	0.25	
U_4	Oil crisis	0.15	
U_5	Euro:Dollar = 1:2	0.01	
E_1	Increase in visiting tourists	0.08	0.10
E_2	Increase in demand for services	0.05	0.11
E_3	Increase in traffic congestion	0.11	0.66
E_4	Agricultural development	0.06	0.19
E_5	Increase in real-estate values	0.09	0.23
E_6	High added-value economic activities	0.06	0.41
E_7	Population decrease, especially young people	0.1	0.75
E_8	Increase in unskilled immigration	0.1	0.30
E_9	Increase in unemployment	0.07	0.34
E_{10}	Industrial development	0.07	0.51

Table 2 The set of actions in the example presented. All actions are defined in the interval [0, 10]

Id	Description
A_1	Financial support for agricultural development
A_2	Better services for citizens
A_3	Improvement in collective transport systems
A_4	Renovation of the historical center
A_5	Extension and foundation of new universities
A_6	Investment in new technologies
A_7	Support for entrepreneurial start-ups
A_8	Measures for environmental protection and territorial preservation
A_9	Better quality high-school education

Table 3 The set of arcs in the example presented. All shape functions are sigmoidal with $\alpha = 0.1$ and $\beta = 0.9$ (see Fig. 5)

Causal influence	Activation value	Causal strength	Causal influence	a_α	a_β	p_0 or p_{inf}
$U_1 \to E_4$	u_1	0.40	$A_1 \to E_4$	5.00	10.00	0.60
$U_4 \to E_{10}$	$\neg u_4$	0.30	$A_2 \to E_5$	2.00	7.00	0.45
$E_1 \to E_5$	e_1	0.40	$A_2 \to E_7$	8.00	3.00	0.40
$E_3 \to E_5$	$\neg e_3$	0.25	$A_2 \to E_1$	2.00	8.00	0.50
$U_2 \to E_4$	$\neg u_2$	0.14	$A_3 \to E_3$	10.00	4.00	0.60
$U_3 \to E_{10}$	$\neg u_3$	0.25	$A_3 \to E_5$	2.00	8.00	0.30
$E_{10} \to E_9$	$\neg e_{10}$	0.40	$A_4 \to E_5$	3.00	8.00	0.40
$E_{10} \to E_6$	e_{10}	0.14	$A_4 \to E_1$	5.00	10.00	0.40
$E_7 \to E_5$	$\neg e_7$	0.25	$A_5 \to E_7$	10.00	5.00	0.40
$E_4 \to E_8$	e_4	0.30	$A_6 \to E_{10}$	2.00	6.00	0.70
$E_4 \to E_9$	$\neg e_4$	0.15	$A_7 \to E_{10}$	2.00	7.00	0.60
$E_1 \to E_2$	e_1	0.20	$A_7 \to E_6$	5.00	10.00	0.40
$E_7 \to E_8$	e_7	0.15	$A_8 \to E_{10}$	10.00	5.00	0.20
$E_9 \to E_7$	$\neg e_9$	0.20	$A_8 \to E_5$	2.00	10.00	0.40
$E_5 \to E_6$	$\neg e_5$	0.35	$A_8 \to E_4$	5.00	10.00	0.40
$U_5 \to E_1$	u_5	0.70	$A_9 \to E_6$	5.00	10.00	0.30

the arcs represents the intensity of the causal strength (i.e. the maximum or the asymptotic value in the case of arcs from actions to events). The model required the estimation of 15 leak probabilities, 16 constant causal strength and 16 shapes of functions, the latter selected via the program graphical interface. Three different objectives were specified, with parents defined as:

– the set of actions $\mathcal{A}$ as $\pi(C_1)$;
– the set $\mathcal{G} \subseteq \mathcal{V}$, as $\pi(C_2)$, containing events reputed positive facts by the panel of domain experts;
– the set $\mathcal{B} \subseteq \mathcal{V}$, as $\pi(C_3)$ of events reputed negative facts;

In particular, the objective functions were defined as:

$$g_{C_1} = \sum_{A_i \in \mathcal{A}}^{m} A_i, \quad g_{C_2} = \frac{1}{\sharp\mathcal{G}} \sum_{X_i \in \mathcal{G}} x_i, \quad g_{C_3} = \frac{1}{\sharp\mathcal{B}} \sum_{X_i \in \mathcal{G}} (1 - x_i) \tag{16}$$

where it is assumed that $x = 1$ and $\neg x = 0$. Thus, $E_\alpha(C_1)$ is simply the total effort spent on the strategy α, $E_\alpha(C_2)$ is the expected share of occurring events in $\mathcal{G}$, while $E_\alpha(C_3)$ is the expected share of non-occurring events in $\mathcal{B}$. It is worthwhile mentioning that once the model has been defined, the program allows the user to define a different set of objectives which corresponds to a different point of view in the analysis.

4.1 Case I: A three-objective problem

The first analysis was carried out with the aim to maximize the expected values of g_{C_2} and g_{C_3} and to minimize g_{C_1}. The parent sets of the objective functions g_{C_2} and g_{C_3} were defined as $\mathcal{G} = \{E_1, E_4, E_6, E_{10}\}$ and $\mathcal{B} = \{E_3, E_7, E_9\}$, respectively. The NSGA-II algorithm was executed using the settings experimented in [41] for a wide range of tests, that is, a crossover probability $p_c = 0.9$ and mutation probability $p_m = 1/\ell$, where $\ell = 12$ is the number of bits in the chromosome. In order to obtain a Pareto set of good quality, experiments have suggested the use of a population of $N = 500$ individuals, each encoding a strategy (i.e. the 9 effort values relative to the available actions).

The objective functions were evaluated performing the BN stochastic sampling procedure for each individual in the population. Given that the adopted GA was of an elitist kind, the values of the objective functions relative to the current Pareto set were conveniently stored from one generation into its successors (i.e. the BN inferences are not re-performed). Computation was simply terminated after 200 generations (the software allows real-time monitoring of the Pareto-set evolution). Using a standard PC, about thirty minutes were required for the total computation.

Figure 7, representing the final non-dominated set composed of about 400 individuals, shows how the proposed multi-objective approach allows the user to pick a solution from a variety of possibilities. Clearly, the final selection must be performed on the basis of some additional subjective decision. The selected strategies in our case are labeled as a and b in the figure. In particular, Fig. 8 reports the effort allocation suggested by the BayMODE analysis for both strategies, while Fig. 9 represents the corresponding probability variations.

For example, according to strategy a, the decision maker should invest high efforts in "Support for entrepreneurial start-ups", "Support for agricultural development" and "Better services for citizens", medium effort in "Investments

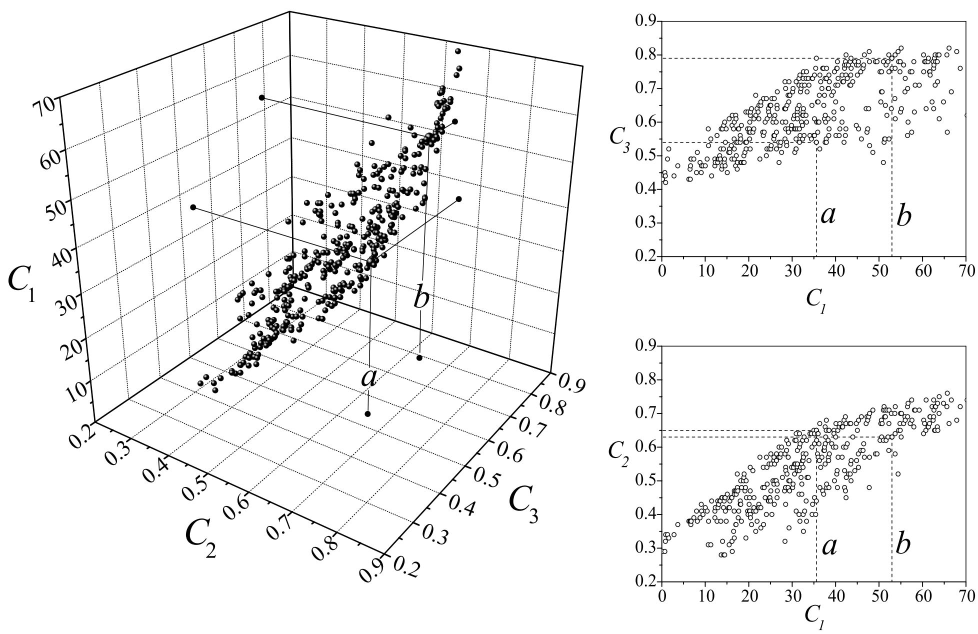

Fig. 7 Case I: the set of computed non-dominated solutions in the space of the objective functions. The selected solutions are labeled

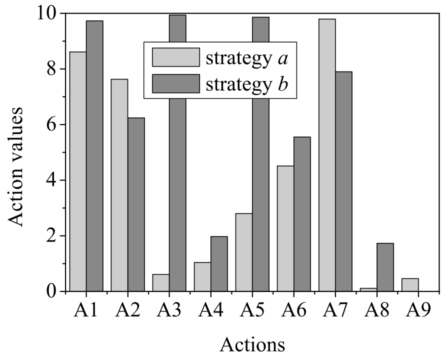

Fig. 8 Case I: effort allocation corresponding to the selected solutions

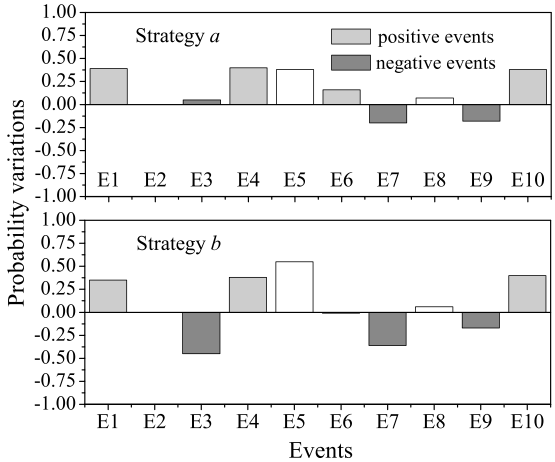

Fig. 9 Case I: Variations of the estimated event probabilities, corresponding to the selected solutions

in new technologies" and low effort in other actions. Undertaking strategy a leads to an expected number of occurred events in $\mathcal{G}$ of 65%, and to an expected number of not-occurred events in $\mathcal{B}$ of 54%. As visible in Fig. 7, if the decision maker is willing to spend more effort, he/she may substantially maintain the value of objective C_2, while increasing the probabilities of C_3: this is the case of strategy b. According to the latter, the decision maker should spend more effort, especially on the actions "Improvement of collective transport systems" and "Extension and foundation of new universities", obtaining as a result an expected number of occurred events in $\mathcal{G}$ of 63%, and an expected number of non-occurred events in $\mathcal{B}$ of 79%.

4.2 Case II: A two-objective problem

In the second analysis, a two-objective problem of maximizing the expected value of g_{C_2} and minimizing g_{C_1} was solved. The parent sets of g_{C_2} were defined as $\mathcal{G} = \{E_5\}$, that is, the problem consisted of maximizing the probability of increasing the real-estate values, while minimizing the total effort. In this case the GA population was composed of $N = 100$ individuals as in [41], while all the other GA parameters were identical to those in Case I. Computation was terminated after 50 generations when the quality of the Pareto-set was judged to be sufficient. Figure 10 represents the final non-dominated set where three selected strategies are labeled with the letters a, b and c. Figure 11 reports the effort allocations suggested by the BayMODE analysis in correspondence to the selected strategies.

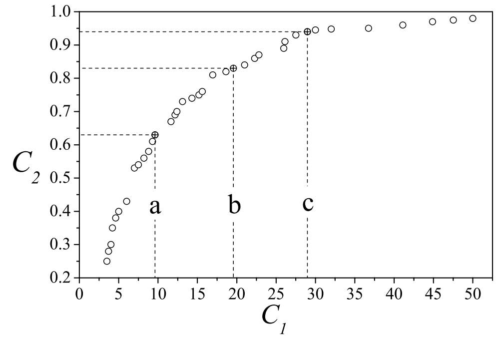

Fig. 10 Case II: the set of computed non-dominated solutions. The selected solutions are labeled

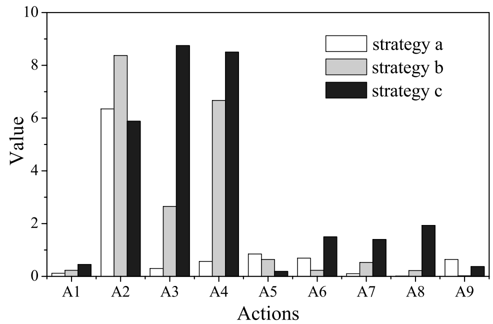

Fig. 11 Case II: effort allocation corresponding to the selected solutions

Strategy a corresponds to a moderate value of the total effort: it suggests that, according to the knowledge provided by the panel of experts who built the model, a decision maker can obtain an increment in the probability of increasing the real-estate values, from $P(e_5) = 0.23$ (see Table 1) to $P(e_5) = 0.63$, simply with high investments in "Better services for citizens" and very low investments in other actions. According to the strategy b, which leads to $P(e_5) = 0.83$, the most important actions are "Better services for citizens" and "Renovation of the historical center", while a moderate investment in "Improvement of collective transport systems" is suggested. Increasing the latter allows a further increment in the value of $P(e_5)$, that in the strategy c is estimated as 0.94.

5 Conclusions and future work

In this paper, we have presented the main features of the decision support system BayMODE, which is based on a simultaneous decision network coupled with a meta heuristic search procedure. The proposed tool includes some characteristics which, put together, make it particularly effective to assist a decision maker who has to achieve a probabilistic trade-off between different objectives, simultaneously undertaking different actions.

The kind of decision network adopted in BayMODE is especially suitable when the noisy-OR gate is chosen to model the interactions between the entities. In fact, in this case, even if actions are represented by non-binary variables, the model requires few parameters, thanks to the use of causal strength functions representing interaction patterns. The latter can easily be defined via the interactive graphical interface. This makes BayMODE particularly useful when the model cannot be automatically obtained from data and must be built by a panel of domain experts. Another significant characteristic of the program is the multi-objective approach, which results in providing the decision maker with a set of non-dominated solutions in a single run of the search algorithm. In applications which include interaction with human decision makers, such an approach can provide a considerable amount of insight that helps in choosing the superior decision strategy or even in assisting in improving the model. Nevertheless, although some systematic method exists [21], it is well-known that in the multi-objective approach, the final choice of the preferred solution might be not easy. This is particularly true when dealing with many objectives and when the number of non-dominated solutions is high. On the contrary, the multi-attribute utility approach is not affected by dimensionality problems.

Further work could be addressed to experimenting different, and possibly more specific, search heuristics, in order to provide non-dominated sets of better quality and with greater computational efficiency. Moreover, in the future we should explore how the definition of the strength functions can be done, integrating experts' opinions with other sources of information.

References

1. Druzdzel MJ, Flynn RR (2000) Decision support systems. In: Encyclopedia of library and information science, vol 67. Marcel Dekker, Inc., New York, pp 120–133
2. von Neumann J, Morgenstern O (1944) The theory of games and economic behavior. Princeton University Press, Princeton
3. Pearl J (1988) Probabilistic reasoning in intelligent systems. Morgan Kaufman, San Mateo, CA
4. Korb, KB, Nicholson AE (2004) Bayesian artificial intelligence. CRC/Chapman and Hall, Boca Raton
5. Murphy K (2006) Software packages for graphical models/Bayesian networks. http://www.cs.ubc.ca/ murphyk/Software/bnsoft.html
6. Morgan MG, Henrion M (1990) Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis. Cambridge University Press, New York
7. Druzdzel MJ (1999) SMILE: structural modeling, inference, and learning engine and GeNIE: a development environment for graphical decision-theoretic models. In: Proceedings of the sixteenth national conference on artificial intelligence and eleventh conference on innovative applications of artificial intelligence, AAAI Press/The MIT Press, Orlando, Florida, USA. 1999, pp 902–903
8. Madsen AL, Jensen F, Kjærulff U, Lang M (2005) The Hugin tool for probabilistic graphical models. Int J Artif Intell Tools 14(3):507–544
9. Haddawy P, Jacobson J, Kahn CE Jr (1997) BANTER: a Bayesian network tutoring shell. Artif Intell Med 10(2):177–200
10. Milho I, Fred ALN (2000) A user-friendly development tool for medical diagnosis based on Bayesian networks. In: Proceedings of the second international conference on enterprise information systems. Stafford, UK, pp 176–180
11. Howard R, Matheson J (1984) Influence diagrams. In: Howard R, Matheson J (eds), The principles and applications of decision analysis. Strategic Decisions Group, Menlo Park, CA, pp 719–762
12. Zagorecki A, Druzdzel MJ (2004) An empirical study of probability elicitation under noisy-OR assumption. In Proceedings of the seventeenth international Florida artificial intelligence research society conference, 2004
13. Diehl M, Haimes YY (2004) Influence diagrams with multiple objectives and tradeoff analysis. IEEE Trans Syst Man Cybern—A 34(3):293–304
14. Pareto V (1896) Cours d'Economie politique, vol I, II. F. Rouge, Lausanne
15. Henrion M (1989) Uncertainty in artificial intelligence 3, ch. Some practical issues in constructing belief networks. Elsevier Science Publishing Company, Inc., pp 161–173
16. Díez FJ (1993) Parameter adjustment in Bayes networks. The generalized noisy-OR gate. In: Proceedings of the ninth annual conference on uncertainty in artificial intelligence. Morgan Kaufmann, pp 99–105
17. Zhang W, Ji Q (2006) A factorization approach to evaluating simultaneous influence diagrams. IEEE Trans Syst, Man Cybern—A 36(4):746–754
18. Keeney R, Raiffa H (1976) Decision with multiple objectives: preferences and value trade-offs. John Wiley & Sons, Inc., New York
19. Evans GE (1984) An overview of techniques for solving multiobjective mathematical programs. Manage Sci 30(11):1268–1282
20. Zeleny M (1982) Multiple criteria decision making. McGraw-Hill, New York
21. Haimes YY, Hall WA (1974) Multiobjectives in water resources systems analysis: the surrogate worth trade-off method. Water Resour Res 10(4):615–624
22. Pearl J (2000) Causality. Cambdrige University Press, New York, NY
23. Shachter RD (1986) Evaluating influence diagrams. Oper Res 34(6):871–882
24. Bellman R (1957) Dynamic programming. Princeton University Press, Princeton
25. Cooper G (1988) A method for using belief networks as influence diagrams. In: Procceedings of the twelfth conference on uncertainty in artificial intelligence. Morgan Kaufmann, pp 55–63
26. Shachter RD, Peot MA (1992) Decision making using probabilistic inference methods. In: Proceedings of the eighth annual conference on uncertainty in artificial intelligence. Morgan Kaufmann, pp 276–283
27. Zhang NL (1998) Probabilistic inference in influence diagrams. In: Proceedings of the fourteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann, pp 514–522
28. Shenoy P (1992) Valuation-based systems for Bayesian decision analysis. Oper Res 40(3):463–484

29. Jensen F, Jensen FV, Dittmer SL (1994) From influence diagrams to junction trees. In: Proceedings of the tenth annual conference on uncertainty in artificial intelligence. Morgan Kaufmann, pp 367–373
30. Zhang NL, Qi R, Poole D (1994) A computational theory of decision networks. Int J Approx Reasoning 11(2):83–158
31. Jensen FV, Vomlelová M (2002) Unconstrained influence diagrams. In: Proceedings of the 18th conference in uncertainty in artificial intelligence. Morgan Kaufmann, pp 234–241
32. Nielsen TD, Jensen FV (1999) Welldefined decision scenarios. In: Proceedings of the fifteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann, pp 502–511
33. Rojas-Guzmán C, Kramer MA (1993) GALGO: A Genetic ALGOrithm decision support tool for complex uncertain systems modeled with Bayesian belief networks. In: Proceedings of the ninth annual conference on uncertainty in artificial intelligence. Morgan Kaufmann, pp 368–375
34. Rojas-Guzmán C, Kramer MA (1996) An evolutionary computing approach to probabilistic reasoning on Bayesian networks. Evol Comput 4(1):57–85
35. Gelsema ES (1995) Abductive reasoning in Bayesian belief networks using a genetic algorithm. Pattern Recognit Lett 16(8):865–871
36. Gelsema ES (1996) Diagnostic reasoning based on a genetic algorithm operating in a Bayesian belief network. Pattern Recognit Lett 17(10):1047–1055
37. de Campos LM, Gámez JA, Moral S (1999) Partial abductive inference in Bayesian belief networks using a genetic algorithm. Pattern Recognit Lett 20(11–13):1211–1217
38. de Campos LM, Gámez JA, Moral S (2002) Partial abductive inference in Bayesian belief networks—an evolutionary computation approach by using problem-specific genetic operators. IEEE Trans Evol Comput 6(2):105–131
39. Coello CC, Veldhuizen DV, Lamont G (2002) Evolutionary algorithms for solving multi-objective problems. Kluwer Academic Publishers
40. Fonseca C, Fleming P (1995) An overview of evolutionary algorithms in multiobjective optimization. Evol Comput 3(1):1–16
41. Deb K, Agrawal S, Pratap A, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
42. Jensen MT (2003) Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. IEEE Trans Evol Comput 7(5):503–515
43. Nojima Y, Narukawa K, Kaige S, Ishibuchi H (2005) Effects of removing overlapping solutions on the performance of the NSGA-II algorithm. In: Evolutionary multi-criterion optimization, third international conference, proceedings, vol 3410 of Lecture Notes in Computer Science, Springer, 2005, pp 341–354
44. Drzadzewski G, Wineberg M (2005) A comparison between dynamic weighted aggregation and NSGA-II for multi-objective evolutionary algorithms. In: Computational intelligence (IASTED international conference on computational intelligence), IASTED/ACTA Press, pp 327–331
45. Goldberg D (1998) Genetic algorithms and evolution strategy in engineering and computer science: Recent advances and industrial applications. Wiley.
46. Fung RM, Chang K-C (1989) Weighing and integrating evidence for stochastic simulation in Bayesian networks. In: Proceedings of the fifth annual conference on uncertainty in artificial intelligence, Morgan Kaufmann, 1989, pp 209–220
47. Shachter RD, Peot MA (1989) Simulation approaches to general probabilistic inference on belief networks. In: Proceedings of the fifth annual conference on uncertainty in artificial intelligence, Morgan Kaufmann, 1989, pp 221–234
48. Henrion M (1986) Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Proceedings of the second annual conference on uncertainty in artificial intelligence. Morgan Kaufmann, 1986, pp 149–164

Ivan Blecic is Assistant Professor of Economic Appraisal and Evaluation at the Faculty of Architecture in Alghero (University of Sassari, Italy) and member of Interuniversity Laboratory of Analysis and Models for Planning (LAMP). He received a Ph.D. in Planning and Public Policies in 2005 from IUAV University of Venice where he has also been a research fellow at the Department of Planning. His current research interests include analysis and modelling for planning, evaluation techniques and modelling, decision support systems and methods for public participation.

Arnaldo Cecchini graduated *cum laude* in Physics at the University of Bologna in 1972. He is Professor of Analysis of Urban Systems at the Faculty of Architecture in Alghero (University of Sassari), Director of the Urban and Environmental Planning Course, Vice-Dean of the Faculty of Architecture in Alghero and Director of the Interuniversity Laboratory of Analysis and Models for Planning - LAMP. He is the author of more than 100 articles and papers published in books and refereed journals and is an expert in techniques of urban analysis and for public participation: simulation, gaming simulation, cellular automata, scenario techniques.

Giuseppe A. Trunfio gained a Ph.D. in Computational Mechanics in 1999 at the University of Calabria, Italy. He has been a research fellow at the Italian National Research Council where he has worked extensively on the application of parallel computing to the simulation of complex systems. He is Assistant Professor of Computer Engineering at the Department of Architecture and Planning of the University of Sassari and his current research interests include decision support, probabilistic models, neural networks, evolutionary computation and cellular automata.