Logic-Based Decision Support for Strategic Environmental Assessment

Marco Gavanelli
ENDIF, Università di Ferrara

Fabrizio Riguzzi
ENDIF, Università di Ferrara

Michela Milano
DEIS - Università di Bologna

Paolo Cagnoli
ARPA Emilia-Romagna

Abstract

Strategic Environmental Assessment is a procedure aimed at introducing systematic assessment of the environmental effects of plans and programs. This procedure is based on the so called coaxial matrices that define dependencies between plan activities (infrastructures, plants, resource extractions, buildings, etc.) and positive and negative environmental impacts, and dependencies between these impacts and environmental receptors. Up to now, this procedure is manually implemented by environmental experts for checking the environmental effects of a given plan or program, but it is never applied during the plan/program construction. A decision support system, based on a clear logic semantics, would be an invaluable tool not only in assessing a single, already defined plan, but also during the planning process in order to produce an optimized, environmentally assessed plan and to study possible alternative scenarios. We propose two approaches to the problem, one based on (Mixed-Integer) Linear Programming and one on Probabilistic Logic Programming that could be, in the future, conveniently merged to exploit the advantages of both. We test the proposed approaches on a real energy plan and we discuss their limitations and advantages.

1 Introduction

Computational Sustainability [6] is a very recent, interdisciplinary research field that aims to apply techniques from computer science, information science, operations research, applied mathematics, and statistics to the problem of balancing environmental, economic, and societal needs for sustainable development.

Among the many possible applications of information technology to sustainable development, decision support systems represent a very important topic. Currently, environmental experts take decisions, perform evaluations and build plans manually, simply relying on experience, with little or no support from automated tools.

We believe the main reason why decision support systems are not widely applied in this field is twofold: first, despite significant advances in algorithmic research, the current state of decision support systems still faces severe difficulties or cannot cope at all with the highly complex structure of sustainability problems. Second, there is a lack of appropriate models for sustainability related applications. These models should be developed in tight collaboration between computer scientists and environmental scientists, economists and biologists that can provide not only models and data, but also feedback on system solutions.

Sustainable development encompasses three pillars: society, economy and the environment. In this paper, we focus particularly on the environment. We address the problem of defining a logic based decision support system for Strategic Environmental Assessment (SEA), a legally enforced procedure aimed to introduce systematic evaluation of the environmental effects of plans and programs. It typically applies to development, waste, transport, energy and land use plans, both regional and local, within the European Union. In this paper, we consider as a case study the assessment of Emilia Romagna regional plans. SEA is based on the so called *coaxial matrices* that quantify dependencies between activities (e.g. infrastructures and plants) contained in a plan and positive and negative environmental impacts (e.g., alteration of woods,

1

water pollution), and dependencies between impacts and environmental receptors (e.g., quantity of CO_2 in the atmosphere).

We propose two alternative approaches: one exploits (Mixed-Integer) Linear Programming and models coaxial matrices as sets of linear equations and inequalities; this is a simple, efficient model, that presumes the available information to be precise, and assumes that influences can be summed up. The second approach is based on Logic Programs with Annotated Disjunction (LPAD) where activities and impacts are combined using the laws of probability. We apply the two approaches on coaxial matrices referring to eleven types of plans that legally require the SEA. Experiments are performed on a real energy regional plan.

The structure of the paper is as follows: section 2 describes regional planning and the Strategic Environmental Assessment along with coaxial matrices. Section 3 recalls the main concepts behind probabilistic logic programming along with its causal interpretation. Section 4 shows the implementation of the coaxial matrices in Linear Programming, while section 5 describes the approach based on LPAD. Section 6 presents experimental results on a real energy plan. A discussion and a description of open issues conclude the paper.

2   Strategic Environmental Assessment

Regional planning is the science of efficient placement of land use activities and infrastructures for the sustainable growth of a region. Our case study is the Emilia Romagna region of Italy, and we developed this work on real data provided by the environmental regional agency. Regional plans are classified into *types*; the SEA is legally required for eleven types of plans (namely Agriculture, Forest, Fishing, Energy, Industry, Transport, Waste, Water, Telecommunication, Tourism, Urban and Environmental plans), those addressed in this work. Each plan defines activities that should be carried out during the plan implementation. Activities are roughly divided into six types:

- infrastructures and plants;

- buildings and land use transformations;

- resource extraction;

- modifications of hydraulic regime;

- industrial transformations;

- environmental management.

Before any implementation, these plans have to be environmentally assessed, under the *Strategic Environmental Assessment Directive*. SEA is a method for incorporating environmental considerations into policies, plans and programmes that is prescribed by European Union policy.

One of the instruments used for assessing a regional plan in Emilia Romagna are the so called *coaxial matrices*, that are a development of the network method [16].

One matrix $\mathcal{M}$ defines the dependencies between the above mentioned activities contained in a plan and positive and negative *impacts* (also called *pressures*) on the environment. Each element m_j^i of the matrix $\mathcal{M}$ defines a qualitative dependency between the activity i and the negative or positive impact j. The dependency can be *high*, *medium*, *low* or *null*. Examples of negative impacts are energy, water and land consumption, variation of water flows, water and air pollution and so on. Examples of positive impacts are reduction of water/air pollution, reduction of greenhouse gas emission, reduction of noise, natural resources saving, creation of new ecosystems and so on.

The second matrix $\mathcal{N}$ defines how the impacts influence environmental receptors. Each element n_j^i of the matrix $\mathcal{N}$ defines a qualitative dependency between the negative or positive impact i and an environmental receptor j. Again the dependency can be *high*, *medium*, *low* or *null*. Examples of environmental receptors are the quality of surface water and groundwater, quality of landscapes, energy availability, wildlife wellness and so on.

The matrices currently used in Emilia Romagna contain 93 activities, 29 negative impacts, 19 positive impacts and 23 receptors and refer to the above mentioned 11 plans.

The coaxial matrices are currently used by environmental experts that manually evaluate a single, already defined, plan. A plan basically defines the so called *magnitude* of each activity: magnitudes are real values that intuitively express *"how much"* of an activity is performed. The unit is different for each activity: for example, for activity *Thermoelectric power plants* the magnitude says how many MW of electric power will be produced by thermoelectric plants, while the magnitude of *Oil/gas/steam pipelines* gives the number of Km of pipes installed. A manual evaluation of alternatives and *what-if* queries are very difficult to consider. In addition, planning is now carried out without a rigorous consideration of environmental aspects contained in the coaxial matrices.

In this paper, we propose two approaches for the design of a decision support system that can be used to assess a single, already defined plan, to evaluate different scenarios during the planning phase or to optimize the definition of land use activities and infrastructures.

In both cases, we convert the qualitative values into real numbers in the interval $[0, 1]$. The environmental expert suggested the values to be 0.25 for *low*, 0.5 for *medium*, and 0.75 for *high*.

The first approach is based on Linear Programming, that is extremely efficient when dealing with linear equations. On the other hand, this approach does not take into consideration the subjective and stochastic nature of the available data: each value in the matrices is simply used as a coefficient in a linear equation, so we assume that positive and negative impacts derived from planned activities can be summed. While in general, impacts can indeed be summed, in some cases a mere summation is not the most realistic relation and more sophisticated combinations should be considered.

For this reason, we also evaluate a Causal Probabilistic Logic Programming approach that is grounded on the well-established theory of probability and causality. The same coefficients are now interpreted as probabilities, that will be combined through the classical rules to provide the likelihood of a given receptor being affected. The price to be paid is a higher computation time. A realistic decision support system should merge the two approaches and this is subject of current research activity.

3 Background

We provide some preliminaries on the two logic-based techniques used in this paper.

3.1 Causal Probabilistic Logic Programming

In this section we first present Probabilistic Logic Programming and then we discuss how to model causation with it.

3.1.1 Probabilistic Logic Programming

The integration of logic and probability has been widely studied in Logic Programming and various languages semantics have been proposed, such as Probabilistic Logic Programs [2], Independent Choice Logic [9], PRISM [14], pD [5], CLP(BN) [13] and ProbLog [3].

Logic Programs with Annotated Disjunctions (LPADs) [17] are particularly suitable for reasoning about causes and effects [18]. They extend logic programs by allowing clauses to be disjunctive and by annotating each atom in the head with a probability. A clause can be causally interpreted by supposing that the truth of the body causes the truth of one of the atoms in the head non-deterministically chosen on the basis of the annotations.

An LPAD theory T consists of a finite set of *annotated disjunctive clauses*. These clauses have the following form

$$(H_1 : \alpha_1) \vee (H_2 : \alpha_2) \vee \ldots \vee (H_h : \alpha_h) : -B_1, B_2, \ldots B_b$$

where the H_is are logical atoms, the B_is are logical literals and the α_is are real numbers in the interval $[0, 1]$ such that $\sum_{i=1}^{h} \alpha_i \leq 1$. If $\sum_{i=1}^{h} \alpha_i < 1$, the head of the clause implicitly contains an extra atom *null* that does not appear in the body of any clause and whose annotation is $1 - \sum_{i=1}^{h} \alpha_i$. If C is the clause above, $H(C, i)$ is H_i, $\alpha(C, i)$ is α_i and *body*(C) is $B_1, B_2, \ldots B_b$.

The semantics of a non-ground theory T is defined through its grounding $g(T)$ and [17] require that $g(T)$ is finite.

An *atomic choice* χ is a triple (C, θ, i) where $C \in T$, θ is a substitution that grounds C and $i \in \{1, \ldots, n\}$ where n is the number of atoms in the head of C. (C, θ, i) means that, for the ground clause $C\theta$, the head $H(C, i) : \alpha(C, i)$ was chosen. A *selection* σ is a set of atomic choices such that for each clause $C\theta$ in $g(T)$ there exists one and only one atomic choice (C, θ, i) in σ. We denote the set of all selections of a program T by $\mathcal{S}_T$.

A selection σ identifies a normal logic program $T_\sigma = \{(H(C, i) : -body(C))\theta | \ (C, \theta, i) \in \sigma\}$ that is called an *instance* of T. A probability distribution is defined over the space of instances by assuming independence among the choices made for each clause, thus the probability P_σ of an instance T_σ is given by $P_\sigma = \prod_{(C,\theta,i) \in \sigma} \alpha(C, i)$.

The meaning of the instances of an LPAD is given by the well-founded semantics. For each instance T_σ, we require that its well-founded model $WF(T_\sigma)$ is total, since we want to model uncertainty only by means of disjunctions.

The probability of a formula Q is given by the sum of the probabilities of the instances in which the formula is true according to the well-founded semantics:

$$P(Q) = \sum_{\sigma \in \mathcal{S}_T, WF(T_\sigma) \models Q} P_\sigma$$

An LPAD T can be translated into a Bayesian network $\beta(T)$ that has a Boolean random variable for each ground atom plus a random variable $choice_{C\theta}$ for each grounding $C\theta$ of each clause C of T.

$choice_{C\theta}$ assumes value $H(C, i)\theta$ with probability $\alpha(C, i)$ if the configuration of its parents makes the body true, while it assumes value *null* with probability 1 if the configuration makes the body false. A ground atom A has as parents all the $choice_{C\theta}$ variables for which A appears in the head of $C\theta$. A assumes value true with probability 1 if one of the parent choice variables assumes value A, otherwise it assumes value false with probability 1.

Various approaches have been proposed for computing the probability of queries from an LPAD. Riguzzi [10] discusses an extension of SLG resolution, called SLGAD, that is able to compute the probability of queries by repeatedly branching on disjunctive clauses. A different approach was taken by Meert et al. [7], where an LPAD is first transformed into its equivalent Bayesian network and then inference is performed on the network using the variable elimination algorithm. Riguzzi [11] presents the `cplint` system that first finds explanations (sets of atomic choices) for queries and then computes the probability by means of Binary Decision Diagrams, as proposed in [3] for the ProbLog language. `cplint` was used in the experiments in Section 6.2 because of its speed [12].

3.1.2 Causal Models

Determining when an event causes another event is very important in many domains, take for example science, medicine, pharmacology or economics. Causality has been widely debated by philosophers and statisticians: often it has been confused with correlation, while they are in fact distinct concepts, since two events may be correlated without one causing the other. Recently, [8] helped to clarify the concept of causation by discussing how to represent causal information and how to perform inference from it. [8] illustrates two types of causal models: causal Bayesian networks and structural equations.

Causal Bayesian networks differ from standard Bayesian networks because the edge from variable X to variable Y means that X is a cause for Y, while in standard Bayesian networks it simply means that there is a statistical dependence.

Pearl [8] proposed an approach for computing the probability of effects of actions and suggested to use the notation $P(y|do(x))$ to indicate the effect of the action of setting the variable X to value x on the event of variable Y taking value y. $P(y|do(x))$ is different from the probability of y given x ($P(y|x)$) because we do not simply observe $X = x$ but we intervene on the model by making sure $X = x$ is true.

The technique proposed in [8] for computing $P(y|do(x))$ consists of removing the parents of X, setting X to x and computing $P(y)$ in the obtained network.

This approach can be applied to probabilistic logic languages that can be translated to Bayesian networks, such as LPADs. In order to compute the probability of a ground atom Y of being true given an intervention that consists of making a ground atom X true from an LPAD T, we need to remove X from the head of

all the clauses that contain it and by adding X as a fact to T. The probability of Y can then be computed from the resulting LPAD T' by using standard inference, i.e., by computing $P(Y)$.

4 Coaxial Matrices in Linear Programming

The coaxial matrices can be simply interpreted as a linear programming model.

More in detail, we consider the values of the magnitudes (see Section 2) as a vector $\mathbf{A} = (a_1, \ldots, a_{N_a})$. Now, the environmental impacts caused by the activity i (with magnitude a_i) can be estimated with the system of linear equations

$$\forall j \in \{1, \ldots, N_p\} \quad p_j = m_j^i a_i.$$

When considering a whole regional plan, we sum up the contributions of all the activities and obtain the estimate of the impact on each environmental impact/pressure:

$$\forall j \in \{1, \ldots, N_p\} \quad p_j = \sum_{i=1}^{N_a} m_j^i a_i. \tag{1}$$

In the same way, given the vector of environmental pressures $\mathbf{P} = (p_1, \ldots, p_{N_p})$, one can estimate the influence on the environmental receptor r_i by means of the matrix $\mathcal{N}$, that relates pressures with receptors:

$$\forall j \in \{1, \ldots, N_r\} \quad r_j = \sum_{i=1}^{N_p} n_j^i p_i. \tag{2}$$

The system of equations (1-2) are imposed as linear constraints; thanks to this formalisation, a number of queries of high interest both for the planner and for the evaluator of the environmental policy can be posed to the system.

The final goal for the evaluator of the environmental policy is computing the environmental footprint of a devised plan. The plan is given as a set of values representing the magnitude of each of the activity. In other words, given the set of values $\mathbf{A} = (a_1, \ldots, a_{N_a})$, we can compute the environmental footprint $\mathbf{R} = (r_1, \ldots, r_{N_r})$, simply by applying equations (1) and (2).

Another query studies the impact of a single unit (in a standardized format) of activity a_i; for example, we are interested to know what is the environmental footprint of producing 1MW of electric power through a thermoelectric plant. We instantiate the vector of activities to a unary vector with $a_i = 1$ if $i = therm$ and $a_i = 0$ otherwise:

$$\mathbf{A} = (0, 0, \ldots, 1, \ldots, 0)$$

In this way, one can find out, by looking at the resulting vector $\mathbf{R}$, which of the receptors are (positively or negatively) influenced by the devised activity. Also, one can get an estimate of those receptors that are more heavily influenced, and those that are only marginally influenced. This query can be used by experts to calibrate the numbers in the coaxial matrices, by considering singularly each activity.

Another important query for the final user is asking which of the possible activities (always in normalized form) has a major impact on some given receptor r_i. In fact, in linear programming one can maximize or minimize some objective function, so the model becomes

$$\begin{array}{cc} & max(r_i) \\ s.t. & (1)(2) \\ & \sum_j a_j = 1 \\ & \forall j, a_j \text{ is integer} \end{array}$$

Finally, if there are laws imposing limits on some receptors (limits for CO_2 emissions, for example) one can very easily impose constraints on receptors (e.g., $r_{CO_2} \leq limit_{CO_2}$), and find if an activity can either be performed at all, or if it requires some compensation (e.g., another activity that improves on the receptor, like reforestation for CO_2), or if it can be done in association with other activities.

In cases where there are two or more alternative activities that cater for the same need, the regulations prescribe that alternatives should be studied, and compared. For example, the need for additional electrical

power is satisfied by building a new plant; however one can choose the type of plant, depending on the environmental conditions. In an area with highly polluted air, a thermoelectric plant could raise the pollution over the law limit, so a different type of plant could be devised, like a solar power plant. On the other hand, a solar plant could be too expensive, and make unaffordable other activities that are necessary in the area (e.g., building a school, a hospital, etc.). In this case, the planner can impose a constraint stating that there is a regional need for at least k MW of electrical power; he/she imposes

$$\sum_{i \in PowerPlants} a_i \geq k$$

(where $PowerPlants$ is the set of indices in the vector $\mathbf{A}$ corresponding to those plants that provide electrical power) and then can optimize for one of the receptors, e.g., r_{CO_2}, or some weighted sum of receptors of interest. Or, the planner may ask what is the maximum power that can be generated in the region without violating the law limits on the receptors

$$
\begin{array}{ll}
& max \sum_{i \in PowerPlants} a_i \\
s.t. & (1)(2) \\
\forall i \in \{1, \ldots, N_r\} & r_i \leq limit_i
\end{array}
$$

In this way, we find the maximum number of MW that can be produced, as well as the electrical power produced by each type of plant. Note that in this way the solver could find an assignment that imposes the execution of compensation activities, as hinted earlier. If there are not enough resources for compensation, we can impose that such activities must not be performed (e.g., by assigning value 0 to all these activities), or we can impose that, given a vector $\mathbf{C}$ with the cost of each activity, the total cost of the activities should not be higher than the allotted finances F:

$$\sum_{i=1}^{N_a} c_i a_i \leq F \tag{3}$$

In the same way, other types of resources, like time, person-months, energy, can be taken into consideration.

We are currently improving the model to take into account the fact that different activities can have different impacts on the environment depending on the type of zone they are placed. For example, if we build a power line within a natural park, its impact is definitely higher than building it near a city. An additional feature we are studying is the fact that depending on the zone we are considering, different impacts might have different weights. For instance, the water quality is extremely important on a river delta, where the whole ecosystem relies on the river water, while it is less important in an industrialized area.

4.1 Sensitivity Analysis

The simplex algorithm provides the optimal value of the objective function, the optimal assignment to the decision variables, and also other information that is of high interest for the decision maker. In particular, it provides the so-called *reduced costs*, and the *dual solution*. These indicators provide precious information on the sensitivity of the found solution to the parameters of the constraint model.

The dual solution is a set of values that correspond to the constraints. It can be thought as the derivatives of the objective function with respect to the right hand side (RHS) of the constraints. This means that we immediately see, in the dual solution, which of the constraints are *tight*, i.e., which would change the value of the objective function if the RHS coefficient changes. For example, if we are optimizing the number f_{MW} of MW of electric power, and we have a constraint $r_{CO_2} \leq limit_{CO_2}$ and the corresponding dual value d_{CO_2} in the optimal solution is non-zero, this means that $d_{CO_2} = \frac{\partial f_{MW}}{\partial limit_{CO_2}}$. In other words, the value of the dual variable d_{CO_2} answers to the question: *"How much would the production of energy decrease in case the limit of CO_2 lowers of one point?"* This is an important information, since regulations change, and tend to become more strict.

The same analysis can be performed on the problem of optimizing some (weighted sum of) receptors, given a total number of plants (or required MW). In this case, the dual value associated to a constraint represents how much the receptor will improve if that constraint is partially relaxed (if the RHS becomes less strict). For example, suppose we are optimizing the emissions of nitrogen oxides (NO_x), and we have

the constraint (3) stating a limit on the total cost of the activities, for example, in euro. After obtaining the optimal value, the planner could ask: *"Suppose now that we had more money: if I add one euro, how much would the emissions of NO_x decrease?"* The answer is the dual value d_e of the constraint (3). This analysis is very attractive for the evaluator.

5   A Causal Model for Coaxial Matrices

In this section we consider an interpretation of Coaxial Matrices that differs from the one in Section 4. Instead of associating a real number to each activity, impact and receptor, we associate a Boolean random variable to each of them and we consider the interaction levels expressed in the matrix as probabilistic causal dependencies. In this approach, we assume that an activity is either carried out or not, an impact is either present or not and a receptor is either achieved or not. In other words, we do not consider the magnitude or level of the variables under analysis. We used this approximation to get useful insights on the probabilistic modeling of the problem. In the future, we plan to consider more refined approximations with multivalued random variables or even continuous random variables.

Activities, impacts and receptors are represented by LPAD atoms (propositions) and the effects of activities on impacts and of impacts on receptors are expressed by means of LPAD rules that represent the Coaxial Matrices.

The model thus contains rules that express the effect of the activities on the environmental pressures (where m_j^i is an element of the $\mathcal{M}$ matrix):

$$pressure_j : m_j^i \text{ :- } activity_i.$$

Also, there are rules expressing the effect of the activities on the positive impacts:

$$positive\_impact_j : m_j^i \text{ :- } activity_i.$$

For example, the model contains the rule

```
'Dispersion of dangerous materials':0.75 :-
  'External movements of dangerous materials'.
```

that relates an activity and a pressure, and the rule

```
'Creation of work opportunities':0.5 :-
  'External movements of dangerous materials'.
```

that relates an activity and a positive impact.

Pressures reduce the probability of receptors, while positive impacts increase it. However adding a clause with a certain atom in the head can only increase the probability of the atom. To model the fact that pressures lower the probability of receptors, we use, for each receptor $receptor_k$, two auxiliary predicates $receptor\_pos_k$ and $receptor\_neg_k$ that collect the evidence in favor or against the achievement of the receptor.

The rules that express the negative effect of the pressures on the receptors take the form:

$$receptor\_neg_k : n_k^j \text{ :- } pressure_j.$$

while the rules that express the positive effect of the positive impacts on the receptors take the form:

$$receptor\_pos_k : n_k^j \text{ :- } positive\_impact_j.$$

where n_k^j is an element of the $\mathcal{N}$ matrix. For example, the rule

```
'Human health/wellbeing_neg':0.25 :-
  'Dispersion of dangerous materials'.
```

expresses a negative effect of a pressure on a receptor, and the rule

```
'Human health/wellbeing_pos':0.75:- 'Creation of work opportunities'.
```

expresses a positive effect of a positive impact on a receptor.

Finally, the positive and negative evidence regarding the receptor are combined with the following rules:

$$
\begin{aligned}
receptor : 0.1 \quad &:- \quad \backslash+\ receptor\_pos, receptor\_neg. \\
receptor : 0.5 \quad &:- \quad \backslash+\ receptor\_pos,\ \backslash+\ receptor\_neg. \\
receptor : 0.5 \quad &:- \quad receptor\_pos, receptor\_neg. \\
receptor : 0.9 \quad &:- \quad receptor\_pos,\ \backslash+\ receptor\_neg.
\end{aligned}
$$

For example, the model contains the rules

```
'Human health/wellbeing':0.1 :- \+ 'Human health/wellbeing_pos',
                                    'Human health/wellbeing_neg'.
'Human health/wellbeing':0.5 :- \+ 'Human health/wellbeing_pos',
                                 \+ 'Human health/wellbeing_neg'.
'Human health/wellbeing':0.5 :-    'Human health/wellbeing_pos',
                                    'Human health/wellbeing_neg'.
'Human health/wellbeing':0.9 :-    'Human health/wellbeing_pos',
                                 \+ 'Human health/wellbeing_neg'.
```

that collect positive and negative effects on the receptor *"Human health/wellbeing"*.

These rules express the fact that *"Human health and wellbeing"* is unlikely if there is no positive evidence on it and there is negative evidence on it (first rule). It is very likely if there is positive evidence on it and no negative evidence on it (last rule). In the other cases, the probability of *"Human health and wellbeing"* is in between (second and third rule).

All the parameters were subjectively estimated and validated by the expert.

6 Experiments

6.1 Experimental results of CLP(R)

The agency for the environment of the Emilia-Romagna region (Italy) kindly provided us with the coaxial matrices used for assessing eleven types of plans (that we translate into the Linear Programming model) and the data of a regional energy plan: for each of the activities, we have a *"magnitude"* value. Thanks to the model described earlier, we are able to compute the corresponding values of pressures and receptors.

Initially the results were counterintuitive: the considered plan concerned energy (aimed at raising the available electrical power in the region), while the receptor *energy availability* had a lower value than the previous year. These types of results may be partially due to the qualitative information contained in the matrices, but also highlight possible human mistakes in the data of the matrices. Indeed, a flaw was found (and fixed) in the matrices, showing how a decision support system can contribute to increase the reliability of the data used for the environmental assessment.

Once the human mistakes had been corrected, we re-run the experiments, and got new results. The new results were highly appreciated by the evaluator: the decision support system foresaw strong decrease of *quality of air* (mainly due to the boost on thermoelectric plants), and *water availability* (since thermoelectric plants need refrigeration).

As the plan had a large impact on some receptors, we tried to improve it from an environmental viewpoint: the magnitude of each of the activities was allowed to deviate up to 1% with respect to the original plan, and we optimized the *quality of air* receptor. We had an improvement of about 20.3% on this receptor, which shows that even by allowing small variations one can get significant improvements. On the other hand, we had a decrease of industrial indicators, such as the *availability of productive resources* or the *availability of energy*.

We also tried two dual experiments. The first considers the given plan, keeps constant all activities except the building of (various types of) power plants, fixes the amount of produced energy, and tries to optimize on the quality of air. The second, instead, maximises the electrical power supply without sacrificing any of the environmental receptors, i.e., none of the receptors could worsen with respect to the original plan. The first experiment gave a positive result: by producing electricity with environmental friendly power plants

(wind-powered aerogenerators) we could produce the same amount of energy but have a 57% improvement on the quality of air.

The second, instead, had a negative result: we could not improve the produced electrical power without worsening at least one receptor. These seemingly contradictory results have actually an interesting explanation. The receptors taken into account by the environmental assessment range on all aspects influenced by a human activity, spanning, e.g., from *value of cultural heritage* to *stability of riverbeds*, from *quality of underground water* to *visual impact on the landscape*. Aerogenerators, recommended in the previous optimization, have a significant visual impact, so they are not implementable unless we relax the visual requirement.

Computing time for this analysis was hardly measurable: all times were far less than a fraction of second with the open-source solver CLP-Cbc [4] on a modern PC. Thanks to such a fast computation, we could to comment on-line the results of the queries with the experts of the regional agency, identify errors in the provided data, and try variations of the parameters.

6.2 Causal Model

Given the causal model presented in Section 5, we can ask various what-if queries

1. if these activities are performed, what is the probability of a certain pressure or of a positive influence of appearing?

2. if these works are performed, what is the probability of a certain receptor being satisfied?

Queries of type 2 are more interesting because they relate the works directly with their final effects of interest. However, they are also more complex to compute. Moreover, the queries above can be generalized to the case in which the activities are performed with a certain probability.

We can answer the queries above by following the approach described in 3.1.2: we add a fact for each activity that is carried out and we ask the probability of the query from the modified program.

We report on a number of queries together with their execution times on Linux machines with an Intel Core 2 Duo E6550 (2333 MHz) processor and 4 GB of RAM.

The probability of the pressure *"Dispersion of Dangerous Materials"* performing the activities *"External movements of dangerous materials"* and *"Internal movements of dangerous materials"* is 0.937500. The CPU time was below 10^{-6} seconds.

The probability of the receptor *"Human health/wellbeing"* given that we perform the activities *"External movements of dangerous materials"* and *"Internal movements of dangerous materials"* is 0.546915 and the query took 22.713 seconds.

If we perform the activity *"Industrial processing and transformation"* the probability of the receptor *"Human health/wellbeing"* is 0.474918, computed in 84.453 seconds. This query takes longer than the previous ones because the work *"Industrial processing and transformation"* has an impact on many more influences than *"External movements of dangerous materials"* and *"Internal movements of dangerous materials"* and all these impacts must be combined to find the effect on *"Human health and wellbeing"*. To give the reader an idea of the complexity of this query, there are 655,660 explanations, 12,847,036 atomic choices appear in the explanations and 42 random variables are involved.

The probability of receptor *"Atmosphere quality, microclimate"* given the action *"Industrial processing and transformation"* is 0.360851. The CPU time was 0.02s.

If we add the activity *"Oil and gas extraction plants"* the probability of the receptor *"Atmosphere quality, microclimate"* lowers to 0.326481, computed in 6.852s

By adding the activity *"Fire extinguishing plants"* the probability of the receptor *"Atmosphere quality, microclimate"* rises to 0.454471, due to the positive effects of the last activity. The CPU time was 92.67 seconds.

As can be seen from the last three cases, increasing the number of activities increases the computation time, since we have to combine the effects of the different causes. The last query has 606,726 explanations, 10,973,022 atomic choices appear in the explanations and 36 random variables are involved.
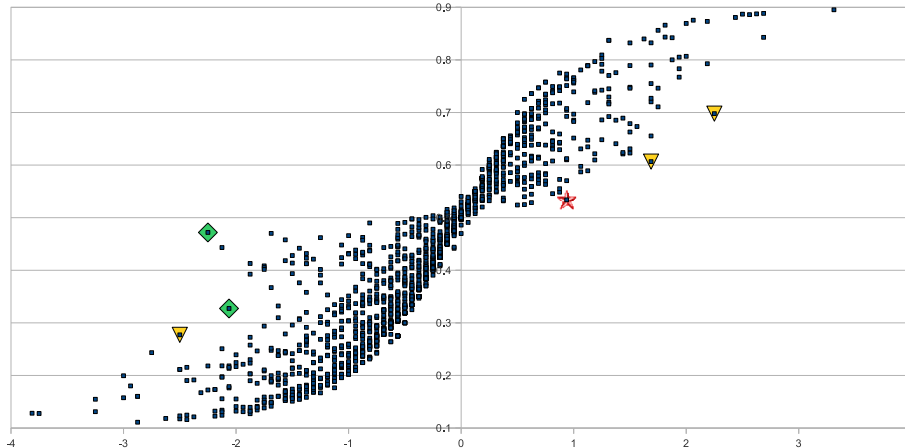
Figure 1: Scatter plot of linear vs probabilistic values

6.3 Comparison

As hinted earlier, we developed two models with the final aim to integrate the two. Before such an ambitious goal can be reached, we need to identify strengths and weaknesses of the two. In order to have a systematic comparison, we produced two tables (one for each approach) in which each cell contains the effect of single activity on a single receptor. One table reports the results of the linear model and the other those of the causal model. The tables are thus of size $N_a \times N_r = 93 \times 23 = 2139$.

In the scatter plot of Figure 1 we draw the results of the causal model against those of the linear model: the linear model results are on the x-axis, while in the causal model results are on the y-axis. As we can see, most of the point are clustered along a simple curve, which seem to indicate a close relationship between the two models that we are going to investigate in the near future. Moreover, whenever the linear value is positive, the probability of improving the receptor is greater than 0.5 and vice-versa, showing that the two models may disagree on the values, but they agree on the direction of the effect on the receptor.

To better investigate the results, we considered the points farthest from the curve that are highlighted in Figure 1. Since points are those for which the linear and causal model differ the most. We asked the expert to evaluate the results for those points, to understand which model gave the best result. For the points shown as a triangle in Figure 1, the expert was unable to state which answer is better. For the points with a diamond symbol, the Linear Programming approach gave a better result. In the point shown as a star, both approaches failed to give a correct result.

From these results, we can say that often the effects can be summed up, although in some cases other combinations could be necessary. In future work, we plan to try other techniques, such as fuzzy logic, and to use probabilities to model the uncertainty of the matrices parameters.

7 Conclusions

The environmental assessment is now becoming a systematic procedure imposed by the laws, and its importance is doomed to increase year after year. In this paper, we proposed how two technologies taken from computational logic can successfully address practical problems of the environmental assessment. The work was conducted using the real data used in previous years for environmental assessment of Emilia Romagna plans.

Linear Programming showed an efficient management of large models and provides useful sensitivity analysis that can be used by planners and evaluators to assess multiple alternative scenarios. The drawback is that contributions from different activities and different impacts are merely summed, while in some cases more sophisticated combinations are required.

On the other hand, the probabilistic model takes into consideration the subjective and stochastic nature of

the provided data, paying the cost of a higher computational effort. In addition, some activity contributions and impacts should indeed be summed, while others are conveniently merged through probability laws.

We believe that computational logics can have a big impact in this field. One future work will be trying to merge the two models into a single component: in CL and AI, formalisms have been proposed to take under a same umbrella constraints and probabilities, like the Valued CSP model [15] or the semiring framework [1].

References

[1] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based constraint satisfaction and optimization. *Journal of the ACM*, 44(2):201–236, 1997.

[2] Evgeny Dantsin. Probabilistic logic programs and their semantics. In Andrei Voronkov, editor, *Proceedings of the 2nd Russian Conference on Logic Programming*, volume 592 of *Lecture Notes in Computer Science*, pages 152–164, Berlin / Heidelberg, 1991. Springer.

[3] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2462–2467, 2007.

[4] John Forrest and Robin Lougee-Heimer. Cbc user guide. `http://www.coin-or.org/Cbc/`, 2005.

[5] Norbert Fuhr. Probabilistic Datalog: Implementing logical information retrieval for advanced applications. *Journal of the American Society for Information Science*, 51(2):95–110, 2000.

[6] C. P. Gomes. Challenges for constraint reasoning and optimization in computational sustainability. In Ian P. Gent, editor, *Int.l Conference on Principles and Practice of Constraint Programming*, volume 5732 of *Lecture Notes in Computer Science*, pages 2–4, Berlin/Heidelberg, 2009. Springer.

[7] W. Meert, J. Struyf, and H. Blockeel. CP-Logic theory inference with contextual variable elimination and comparison to BDD based inference methods. In L. De Raedt, editor, *Proceedings of the 19th International Conference on Inductive Logic Programming*, Leuven, Belgium, 2009. KU Leuven.

[8] J. Pearl. *Causality*. Cambridge University Press, 2000.

[9] David Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artif. Intell.*, 94(1-2):7–56, 1997.

[10] F. Riguzzi. Inference with logic programs with annotated disjunctions under the well founded semantics. In Maria Garcia de la Banda and Enrico Pontelli, editors, *International Conference on Logic Programming*, number 5366 in Lecture Notes in Computer Science, pages 667–771, Berlin/Heidelberg, 2008. Springer.

[11] Fabrizio Riguzzi. A top down interpreter for LPAD and CP-logic. In R. Basili and M. T. Pazienza, editors, *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, number 4733 in Lecture Notes in Artificial Intelligence, pages 109–120, Berlin/Heidelberg, 2007. Springer-Verlag.

[12] Fabrizio Riguzzi. Extended semantics and inference for the Independent Choice Logic. *Logic Journal of the IGPL*, 17(6):589–629, 2009.

[13] V. Santos Costa, D. Page, M. Qazi, and J. Cussens. CLP($\mathcal{BN}$): Constraint logic programming for probabilistic knowledge. In *Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2003.

[14] Taisuke Sato and Yoshitaka Kameya. PRISM: A language for symbolic-statistical modeling. In Martha E. Pollack, editor, *International Joint Conference on Artificial Intelligence*, pages 1330–1339. Morgan Kaufmann, 1997.

[15] Thomas Schiex, Hélène Fargier, and Gérard Verfaillie. Valued constraint satisfaction problems: Hard and easy problems. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95*, volume 1, pages 631–639. Morgan Kaufmann, 1995.

[16] Jens C. Sorensen and Mitchell L. Moss. Procedures and programs to assist in the impact statement process. Technical report, Univ. of California, Berkely, 1973.

[17] J. Vennekens, S. Verbaeten, and M. Bruynooghe. Logic programs with annotated disjunctions. In Bart Demoen and Vladimir Lifschitz, editors, *International Conference on Logic Programming*, volume 3131 of *Lecture Notes in Computer Science*, pages 195–209, Heidelberg, Germany, 2004. Springer.

[18] Joost Vennekens, Marc Denecker, and Maurice Bruynooghe. CP-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming*, 9(3):245–308, 2009.