A Framework to Model and Manipulate Constraints for Over-constrained Geographic Applications

JAZIRI Wassim
Miracl Laboratory
ISIMS: Higher Institute of Computer Science and Multimedia of Sfax
Pole Technologique Sakiet Ezzit
3021, Sfax, Tunisia.
E-mail: wassim.jaziri@isimsf.rnu.tn

MAINGUENAUD Michel
LITIS Laboratory
Institut National des Sciences Appliquées de Rouen,
Avenue de l'Université, BP 08, F76801 Saint Etienne du Rouvray Cedex, France.
E-mail: michel.mainguenaud@insa-rouen.fr

Abstract

Geographic applications are often over-constrained because of the stakeholders' multiple requirements and the various spatial, alphanumeric and temporal constraints to be satisfied. In most cases, solving over-constrained problems is based on the relaxation of some constraints according to values of preferences. This article proposes the modelling and the management of constraints in order to provide a framework to integrate stakeholders in the expression and the relaxation of their constraints. Three families of constraints are defined: static vs. dynamic, intra-entity vs. inter-entities and intra-instance vs. inter-instances. Constraints are modelled from two points of view: system with the complexity in time of the different involved operators and user with stakeholders' preferences. The methodology of constraints relaxation is based on primitive, complex and derived operations. These operations allow a modification of the constraints in order to provide a relevant solution to a simulation. The developed system was applied to reduce the streaming/floods risks in the territory of Pays de Caux (Seine Maritime, France).

Keywords: Geographic Information Systems, Over-constrained problems, stakeholders' constraints, Constraints relaxation and manipulation, Geographic applications.

1. Introduction

Regional planning applications require geographic data such as maps, aerial photos and treatments such as spatial analysis, path evaluations. In front of the tremendous set of available data, stakeholders are tempted to define numerous constraints to be satisfied within a project. These constraints may be based on specific spatial locations involving spatial analysis (e.g., due to soil erosion, I would like a new distribution of the different crops in a specific part of a regional area without modifying the farmers' incomes) or network oriented operators (e.g., the path, the cows must follow to join the pasture lands, must not cross an important road).

Stakeholders require therefore multiple and complex constraints that make difficult to find relevant solutions. In most works dealing with constraints satisfaction or decision aid, the resolution of over-constrained problems depends on the analyst's point of view and expertise. An over-constrained problem is defined such as "the system of constraints does not allow finding an assessment that satisfies all constraints".
The relaxation of constraints and the aggregation of preferences are treated in numerous works [5], [18], [29]. We provide in this article a framework to describe (i.e., proposition of a generic structure to model the constraints) and manipulate stakeholders' constraints (i.e., proposition of a mechanism for constraints relaxation). This framework aims at allowing the discussion in order to define the relaxation of these constraints. It is based on the modelling and the manipulation of constraints with a set of primitive, complex and derived operations. The applicative context concerns the management of streaming risk in agricultural territories. We define the risk in this application as "the quantity of water that flows on the parcels of a watershed".

Part 2 presents the previous works in the constraints management domain. Part 3 presents the position of our work. Part 4 presents an analysis of constraints in geographic applications. Part 5 presents the constraints modelling. Part 6 presents the constraints manipulation. Part 7 presents the results of experimentations before concluding with the perspectives of this work.

2. Previous works

Constraints manipulations rely on several formal representations and resolution mechanisms. Geographic applications are also concerned with the definition of constraints in order to provide relevant and acceptable solutions.
In this part, we firstly present formal frameworks for constraints manipulations and then geographic applications in this domain.

2.1. Formal frameworks

Formal representations of constraints modelling are mainly developed in the following disciplines: optimisation, constraints satisfaction and decision aid.

2.1.1 Optimization

Optimization consists in finding the best solution in relation with criteria, and in satisfying some constraints. It is often expressed via a mathematical function reflecting the objectives of the problem. The optimization function usually comprises one criterion that determines the optimum. Otherwise, the objective consists in finding a good compromise between multiple criteria [41].
In practice, optimization problems can reach a high complexity and require considerable computation times because of the number of potential solutions. The approximate methods are generally used to resolve this class of problems. The satisfaction of constraints is not the essential task. In this kind of optimization providing a solution is already a challenge therefore in a compromise time/solution the satisfaction generally cannot be reached. Constraints are a guide more than an objective. Stakeholder is willing to relax some constraints for the improvement of the objective. Approximate methods are based on an

iterative exploration of the search space to find a solution (good quality vs. a reasonable computation time). Among the most known approximate methods, we mention the neighbourhood methods, such as local search [33], simulated annealing [16], threshold algorithms [21], noising method [10] and Tabu search [30] as well as the algorithms based on the evolution approach [22] [44].

The optimization methods are inefficient in case of multiple and hard constraints or facing decision problems that do not verify the conditions of optimization. These conditions are the mutual exclusiveness, the exhaustiveness and the complete pre-order [39]. Generally, the optimization methods are limited to large size or under constrained problems. However, real problems often comprise important constraints whose violation can provide inapplicable solutions. The constraints satisfaction problems seem more efficient facing this class of problems which comprise important constraints.

2.1.2 Constraints satisfaction

This framework deals with the problems involving hard constraints to satisfy. It consists in finding one or more solutions satisfying all the constraints. A Constraints Satisfaction Problem (CSP) is defined by a set of variables. Each one may take a value from its domain [19]. Variables involved in a constraint are called linked (by this constraint). Linked constraints represent the conditions to satisfy and restrict the set of values that can simultaneously be affected to the linked variables. The framework of CSP is often used to seek a solution, the best solution or all the solutions of a problem using exact methods.
Among the exact methods, we mention the traditional methods of constraints programming, such as linear programming [32], dynamic programming [15], a number of approaches based on the construction of solutions, such as branch and bound method [34] and several other algorithms based on the backtrack technique [43].

However, the formalism of CSP, better adapted to over-constrained problems, does not pay sufficient importance to the modelling of the constraints and the integration of stakeholders in the decision process. Constraints are supposed to be initially clearly defined, which is not often the case. The decision aid allows the integration of stakeholders in the problem solving.

2.1.3 Decision aid

The decision aid consists in helping decision-makers (stakeholders) to organize and to synthesize their information [42]. Essential terms, describing a decision problem, are alternatives and actors.

- The alternatives indicate the set of possibilities, solutions or possible actions concerned with a decision-making process.

- An actor is an individual (or a group of individuals) that directly or indirectly influences the decision-making. Two types of actors can be distinguished: the decision-maker and the analyst. A decision-maker is an individual (or a group of individuals) who is concerned with the proposed solutions. He is in charge of the expression of the constraints and the evaluation of the different alternatives. An analyst is in charge of modelling the decision problem and of concluding the different phases of a problem solving.

Initial resolution methods belong to Operational Research whose objective consists in solving known and mastered problems [39]. These methods are inefficient facing the problems containing several objectives with multiple aspects such as economic (optimization of profit, of cost), social (adherence, impact on the society), moral (satisfaction) etc. Various methods of decision aid are designed to extend traditional methods (such as linear programming) and to face this type of problems. These methods can be classified according to their evaluation of potential solutions: we distinguish methods without compensation and compensation methods. The first type of methods does not allow any compensation between the different criteria, i.e., they do not accept degraded criteria. The second type of methods authorizes the compensation within criteria (bad values vs. good values of the others) [39].

2.2. Geographic applications

Studies in several disciplines have been initiated to model and conceive solutions for natural risks problems.

In an agricultural application, Martin [35] studies several solutions to reduce the soil erosion due to overland water flow (streaming risk) in a watershed. The developed approach consists in re-affecting crops to the parcels. The technique consists in exploring possibilities of crops succession (rotations on the same parcels) in order to minimize the streaming risk in the most sensitive zones. The author proposes interventions at the level of a parcel (fragmentation of parcels, implantation of inter-crops) and of a farm (another organization of work and manpower). The agronomic constraints of allotment related to the conditions of slope, size and type of required soil for each crop are not taken into account in this study. Other studies are available [8], [14], [36], [31], [38], [45], but the suggested solutions do not take into account all constraints of the problem. In addition, sometimes, farmers are not integrated in the decision making process and/or may disagree with suggested solutions.

In a simulation context, regional development problems have already been treated with various approaches. Some of these approaches were inspired from constraints resolution models, as Le Ber et al. [27], [28] did. These authors relied on the eco-resolution model [17] to solve the problem of crop organization on an agricultural land in order to reach a quantity of production for each crop. They compare the obtained results using a multi-agent system, an expert system and a simulated annealing approach on the watershed of Lignéville (228 parcels) and Valleroy (210 parcels) located on the Région Lorraine (France). However, the authors did not take into consideration a global objective of risk minimization. In [23], the author proposes a multi-agent model to optimize the distribution of crops on agricultural parcels in order to minimize the streaming risk. The evaluation of the streaming risk is performed on the level of the digital terrain model using a cellular automaton. Other steps are done within the framework of multi-agent paradigm [3], the system simulation [2], [11], [25], the constraint satisfaction problems [9], [13], the optimization [24] and the participative approaches [1], [26]. From the participative approaches, the companion modelling [4] has shown its interest in various operational situations, notably in the management of renewable and environmental resources. The companion modelling is based on the dialogue and the interactions between the stakeholders, using a process of collective training. Various simulation tools are developed to facilitate the collective processes of decision, based on the modelling of relationships between natural and social factors [7].

The constraints constitute a common element between these approaches. They determine the applicable methods for a problem solving. They also limit, within an implicit or explicit manner, the analyst's margin and the quality of a solution that can be found for a given problem. However, it seems that a majority of works do not pay sufficient attention to the modelling of the constraints. They are more interested in the choice and the development of adequate methods to satisfy the required constraints.

3. Formal position of this work

Using geographic data, risk management applications can rely on several works in data modelling. We first rely on, in part 3.1, the Open Geospatial Consortium (OGC) specifications to define the service level. Managing constraints in geographic applications requires the cooperation of several techniques we present in part 3.2. The service level relies on a system architecture we present in part 3.3.

3.1. Geographic data

We start from the abstract specification defined in the OGC Abstract Specification (http://www.opengeospatial.org). The OGC specification is designed to enable interoperability among heterogeneous GIS systems. The goal is the full integration of geospatial data and geoprocessing resources into mainstream computing.
The concept of OGC is not concerned with the structure. Its main interest is data behaviour that depends on how components of standards fit together. The Essential Model represents the conceptual interface with the real world. This level is our relevant working level to define users' specifications. To this point of reflection, physical representations of stored data are not taken into account, like Abstract Data Types (ADT) do. An ADT allows manipulating data without any notion of physical representations. As examples of data, we manage in our application:

- A Digital Terrain Model: to provide the circulation of water within the different parcels. It defines the watershed.
- A Land Cover: based on a large set of different crops (e.g., wheat, beet, corn) and the infrastructures (e.g., roads, farms).
- An historical representation of rains within a given period of time.

Any element that may be involved in a constraint in our system (e.g., a parcel, a watershed, a road) is based on the concept of entity and has a set of attributes:

- Alphanumeric attributes to define the semantics of the feature.
- Spatial attributes to define a spatial representation. We take advantage of ADT to be free from the spatial physical model.

Attributes are used to define the constraints that the solution must respect. These attributes are also used to characterize the operators (e.g., the path, the cows must follow to join the pasture lands, must not cross an important road).
OGC is not concerned with the resolution of over-constrained problems.

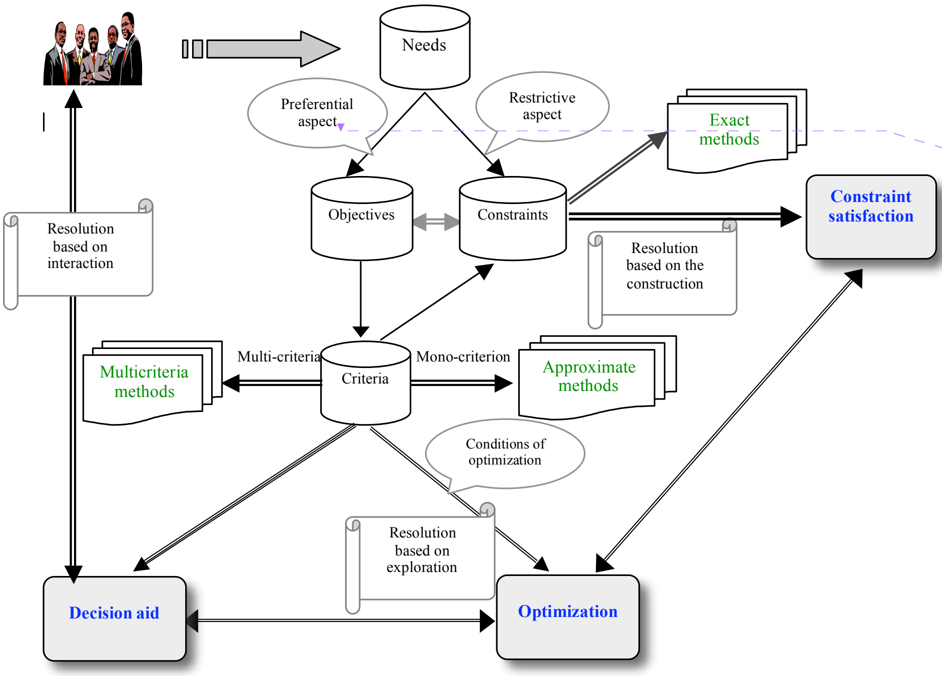
We consider in this article a service application based on 3-tier architecture as a decision aid application (User Interface Manipulation, Constraint Management System, Geographic Database System). Data manipulations can be defined within two levels:

- A service level (here the constraint manipulations).
- A data type level with conventional manipulations defined in the feature Collection Catalogue Services.

In this article, we are only concerned with the service level. Nevertheless, this level uses the access functions of data type (in order to define the complexity in time as a metadata). The relevant operators are manipulations such as, for example, spatial intersection, buffering, adjacency or path evaluations.

3.2. Different techniques

The survey of the optimization, the constraints satisfaction and the decision aid shows that these three disciplines are more complementary than concurrent. They often use the same resolution methods and can be used together (figure 1).



jaziri 19/1/06 16:01
Supprimé: restrictif

Figure 1. *Interactions between disciplines.*

Figure 1 presents the interactions between the optimization, the constraints satisfaction and the decision aid as well as the complementarities between the approximate, exact and multi-criteria methods for problem solving. Single arrows represent the direction of information flow. Double arrows represent a collaborative aspect. The stakeholders express their needs regarding the system, according to two aspects: a preferential aspect which reflects the objectives to reach and a restrictive aspect related to the constraints to satisfy. In case of multiple and hard constraints to satisfy, the problem should be modelled as a Constraints Satisfaction Problem (CSP). The problem generally consists in progressively constructing a solution using exact methods. Otherwise, the satisfaction of conditions of optimization determines two possible frameworks for problem representation and solving: optimization or decision aid. The objectives, expressed by stakeholders, can be formulated according to one or multiple criteria, which, mathematically, are differently expressed. If a single criterion expresses the objectives to reach, the problem is then mono-criteria oriented. Otherwise, it is multi-criteria oriented. According to the mono or multi-criteria nature of a problem, two types of methods (approximate or multi-criteria methods) can be used to solve it. Other types of methods related to the social enquiry can also be used to address decision aid problems, such as the institutional analysis and the social research methods [12].

Regarding decisions-makers' constraints, a traditional problem is to determine whether a set of constraints is coherent or not. As this problem is largely treated in literature, we consider that the set of constraints is coherent. The problem consists in determining whether the problem has a solution or not. We suppose that the number of constraints expressed by the stakeholders is important and therefore transforms our problem into an over-constraint problem. Our goal consists in offering an environment of constraints specification and manipulation to transform an over-constrained problem into a less constrained one.

3.3. System architecture

We propose a decision aid system based on a progressive and generic approach that integrates the stakeholders in the expression, the negotiation and the satisfaction of their needs (constraints). We aim to sensitize stakeholders to collective problems and to lead them to take into consideration flood risks in their agricultural practices. We developed the lower layers of such a system. To validate this architecture, we did not pay a lot of attention to the Human Computer Interaction since several propositions [20], [37] have been provided to define the interactions at the user interface level. We provided a minimal User Interface and we focus on the framework in this article. Figure 2 gives the architecture of such a system.

Constraints are modelled according to a specific grammar. It is complex and difficult to simultaneously satisfy the defined constraints. The system integrates an interaction to negotiate the constraints. This negotiation consists in retaining a subset of constraints with a reasonable complexity, to be treated (i.e., an operational graph that can be provided to a resolution model). The optimization and satisfaction process integrates the minimization of streaming risk and the satisfaction of stakeholders' constraints. The proposed solutions depend on stakeholders' constraints and the required treatments to satisfy them.

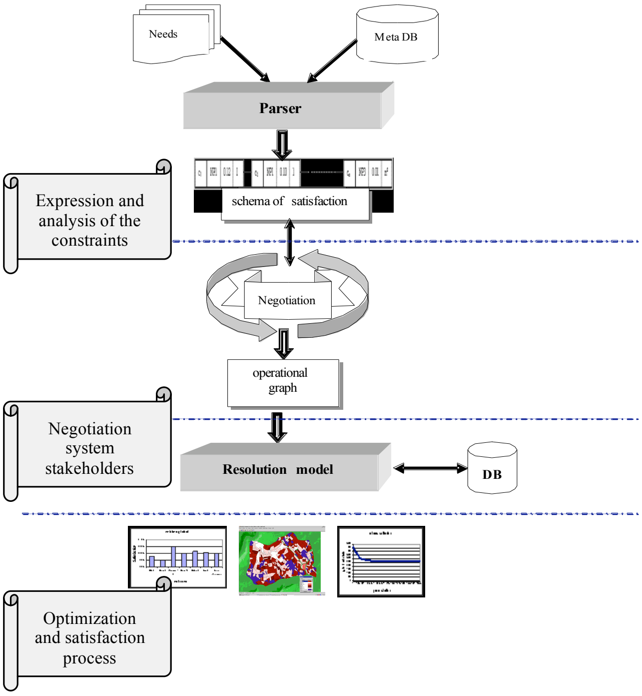


Figure 2. *Structure of a decision aid system.*

4. Analysis of constraints in geographic applications

In this part, we propose an analysis of the various constraints that occur in geographic applications. This analysis is conceptual and therefore application-independent. We illustrate these constraints with some examples extracted from the application we used to validate the approach. We consider a database schema is available to model the different entities (see 3.1) involved in the simulation (e.g., parcels, cultures, a digital terrain model).
We define the concepts used to classify a constraint. The classification of a constraint is a combination of such concepts. The concepts are: static vs. dynamic, intra-entity vs. inter-entities, intra-instance vs. inter-instances.

4.1. Static vs. dynamic

Static and dynamic constraints are based on the following definition:

Definition: A constraint is classified as static (respectively dynamic) if the values of its parameters cannot (respectively can) be changed during the simulation.

As an example, Constraint (1) is defined as: « Flax requires a slope less than 10% ». This value (10%) cannot be changed for physical reasons, therefore it is classified as « static ». Constraint (2) is defined as: « Time allowed for the simulation must not be greater than two hours ». This value can be changed during the various simulations, therefore it is classified as « dynamic ».

This classification defines the analyst's latitude during the simulation. Static constraints represent non-negotiable elements during the search of a solution.

4.2. Intra-entity vs. inter-entities

Intra-entity and inter-entities constraints are based on the following definition:

Definition: A constraint is classified as intra-entity (respectively inter-entities) if its evaluation requires the use of a single (respectively multiple) conceptual entity.

As an example, Constraint (3) is defined as: «Corn: requires the absence of Corn production (a pro-runoff crop) in sensitive downstream parcels of the territory ». This constraint should be respected for each instance of an entity parcel, therefore it is classified as « intra-entity ». Constraint (4) is defined as: « Improvement of the flow risk should be more than 10% for a specific set of parcels ». Several entities are involved in the evaluation of this constraint: wood, parcels, and roads depending on the definition of the flow risk function.

The classification of such constraints involves consequences on the time complexity since an inter-entities requires to manipulate a set of instances instead of a single one (i.e., the selection operator vs. the join operator complexity in the relational algebra).

4.3. Intra-instance vs. inter-instances

Intra-instance and inter-instances constraints are based on the following definition:

Definition: A constraint is classified as intra-instance (respectively inter-instances) if its evaluation requires a unique (respectively several) instance(s) of a conceptual entity.

As an example, Constraint (1) is an intra-instance constraint. Constraint (5) is defined as: «To promote the incomes, the crop re-affectation must provide at least the same incomes as before the re-affectation» is an inter-instance constraint. Several instances of parcels are involved in this constraint in order to evaluate the incomes.

The classification of such constraints involves consequences on the time complexity since an inter-instances constraint requires manipulating a set of instances instead of a single one (i.e., a direct access vs. a selection operator complexity in the relation algebra).

5. Constraint modelling

The choices and requirements are expressed according to a defined syntax. This is formally performed by means of a language with a generative grammar (a formal grammar is a terminal vocabulary, a set of variables, an axiom and a set of production rules).
During the search of a relevant solution, the constraints satisfaction methods use heuristics to choose a couple (variable, value) to be affected [6], [19]. These methods are rarely interested in the order of the constraints to satisfy during the resolution.
Constraint modelling is based on the parameters involved in a constraint and the structures used to model a constraint.

5.1. Constraint parameters

A constraint is represented with five parameters: its identifier, its textual specification, its value of preference, its complexity and its application relevance. The last three parameters represent the user, system and application weight parameters that are used to determine the order of the constraints in the schema of satisfaction. We negotiate these constraints with the stakeholders, using constraints manipulation operators. The retained constraints form the operational graph of constraints (dependent of the application) to be satisfied in the resolution process. We thus advantage the satisfaction of the most relevant constraints according to the operational graph. A decision process involves three components/actors: the user who requires some constraints, the system that must satisfy the constraints and the application weight that expresses the environment of a constraint. We take into account the three aspects and we characterize in the following a constraint by three parameters: a user parameter, a system parameter and an application weight parameter.

- *User parameter* related to the Level of Preference of a constraint (for the user): this parameter reflects the importance of each constraint for a stakeholder and the necessity to be satisfied. The constraints are classified by hierarchical level of preference in order to favour the satisfaction of the most preferred constraints. In real applications, the constraints to satisfy do not generally have the same importance. Works about constraints satisfaction problems distinguish in particular hard constraints that should absolutely be satisfied, and flexible constraints that it is preferable to satisfy [40]. In our context of risk management, we introduce a third level of preference. We consider that the stakeholders require at the same time hard constraints, flexible constraints and secondary constraints. The hard constraints express obligations and must imperatively be satisfied (e.g., Constraint (1)). The flexible constraints represent preferences and are to be satisfied as well as possible. The dissatisfaction of the secondary constraints little reduces the quality (from the user's point of view) of the proposed solution.

- *System parameter* related to the level of computational complexity of a constraint (for the system): this parameter reflects the temporal complexity of the constraints. It thus allows classifying the constraints by increasing order of their complexity in order to treat in first fewer complex one. The complexity of a constraint can be expressed according to the number of involved entities in the database schema, the number of instances of each entity and the number of implied instances at a given instant. This complexity depends on the key or the pseudo-key (i.e., an attribute with an index) of the entities implied in each defined constraint. For each entity implied in the constraint, if the key is given, then an access is an operation of indexation relative to a constant complexity. Otherwise (the key is not given), an access to the required instance is an operation of selection that corresponds to a linear search complexity (this operation requires to traverse (all) instances of an entity to find the required instance in the worst case).

- *Application weight parameter* related to the relevance of a constraint (for the domain expert): this parameter expresses the representative weight of each constraint and its degree of relevance in relation to the application. It is determined by agronomic experts and depends on the type of constraints. We generally distinguish spatial, temporal and alphanumeric constraints. For example, in a context of risk management, and in order to assign crops on the territory parcels, the spatial constraints have more important relevance than the temporal constraints because of the importance of the spatial location of parcels in this spatial process.

5.2. Constraint structures

To represent stakeholders' constraints and the different parameters (user, system and application weight), we use a formalism based on UML (Unified Modeling Language). To simplify the presentation, we do not consider here the modelling of private/public representations and the list of methods (since they are not useful for the presentation).

We consider satisfactionSchema (Figure 3) reflecting the schema of satisfaction of stakeholders' constraints. It represents the core of the constraints from the user's point of view (induced from the user parameter) of three Levels of Preferences (LP) relative respectively to the hard (LP1), flexible (LP2) and secondary (LP3) constraints (from the most important to the least one):

satisfactionSchema
LP1: LP
LP2: LP
LP3: LP

Figure 3. *Representation of satisfactionSchema.*

Each LP (Figure 4) represents three levels of complexity: "`complexityLevel`" (induced from the system parameter). A `complexityLevel` expresses the complexity of evaluation (Φ) in a number of operations in relation to the number of manipulated data. We distinguish the levels of constant (complexity Φ=1), linear ($1<\Phi\leq n$) and stilted complexity ($\Phi>n$):

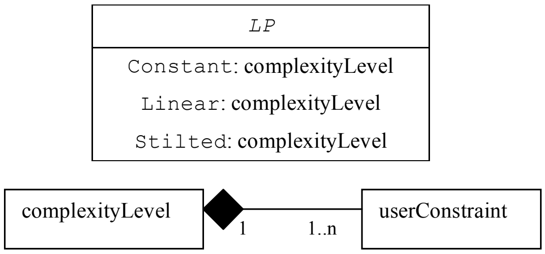


Figure 4. *Representation of the LP (Level of Preference) and complexityLevel.*

SatisfactionSchema (Figure 3) thus represents three levels of complexity for each level of preferences (Figure 4). Each level of complexity (complexityLevel) is a non-ordered collection (set) of stakeholders' constraints. The introduction of the criterion of application relevance (induced from the application weight parameter) reorders the constraints inside each level of complexity. The most relevant constraints are at the head of the rank. Each level of complexity thus represents a list of ordered constraints (i.e., decreasing order) according to their application relevance.

We formally represent a userConstraint (Figure 5) by five attributes, according to this formalism.

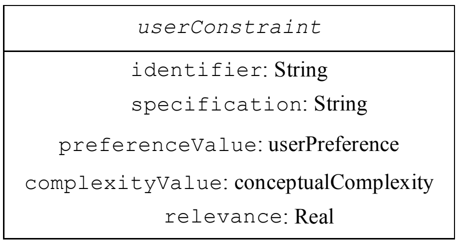


Figure 5. *Representation of userConstraint.*

The `userPreference` and `conceptualComplexity` represent enumerated types (sequence of values indicated by constants of type) defined as follows:

`userPreference = {HARD, FLEXIBLE, SECONDARY}` with `HARD`, `FLEXIBLE` and `SECONDARY` are constants of the type `userPreference`.

`conceptualComplexity = {CONSTANT, LINEAR, STILTED}` with `CONSTANT`, `LINEAR` and `STILTED` are constants of the type `conceptualComplexity`.

The schema of satisfaction corresponds to a list of constraints ordered according to the application relevance. It can be assimilated to a list of ordered constraints according to the

value of preference, the value of complexity and the application relevance, and thus defines a total order (Figure 6):
`satisfactionSchema` $= \langle c^{i}_{jk} \rangle$ where c^{i}_{jk} is the constraint belonging to the level of preference *i* (LPi), of level of complexity *j* and rank *k* (decreasing order according to the application relevance) in the level of complexity *j*.

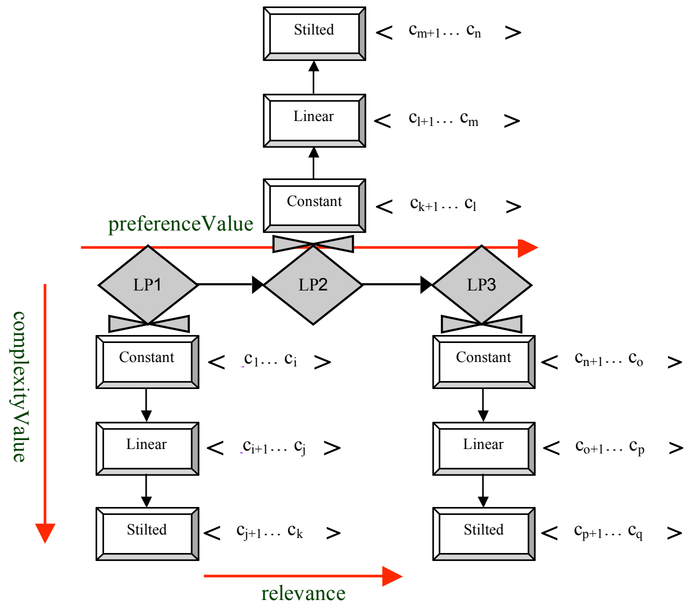


Figure 6. *satisfactionSchema structure.*

This schema of satisfaction reflects the stakeholders' constraints that are ordered according to three parameters: user, system and application weight. It allows establishing an order of constraints satisfaction. This order can be changed using the operators of manipulation.

6. Constraint manipulations

The search of a compromise taking into account multiple constraints and the analyst's margin imposes a dialogue and collaborations between implied actors.

6.1. Principle of negotiation

The stakeholders can control and modify the list of constraints (by means of operations) and visualize the consequences of their choices. This negotiation aims at preserving a list of constraints with a reasonable complexity granting a sufficient margin to the analyst. It consists in assisting and helping stakeholders to relax some constraints rather than relaxing the constraints for them.

To satisfy all stakeholders' constraints, the analyst must have sufficient resources for their satisfaction. These resources can be given using the parameter of complexity. This parameter expresses the cost for the satisfaction of each constraint and thus reflects the temporal complexity. The stakeholders lead a negotiation to find a compromise equilibrating the ratio: response time authorized by the stakeholders (resources of the system) / required time to give solutions (total cost of the constraints). To equilibrate this ratio, stakeholders must modify their constraints and/or the authorized response time to provide a solution (e.g., Constraint 2).

Stakeholders are involved at the beginning of the process to express their constraints and preferences. From this set of constraints, the system evaluates the temporal complexity of the constraints and visualizes this cost to the stakeholders. Several operations on constraints are defined in order to allow stakeholders to reduce the constraints to be satisfied or to increase the authorized response time. The main operations consist in modifying the preferences on a set of constraints or in deleting some constraints according to a criterion (e.g., their relevance, preference or complexity).

6.2. Operators

In addition to conventional operators of manipulation: createList (), listEmpty (), searchElement (), etc., the basic set-oriented operators as the union and the intersection as well as the logic connectors: $\wedge$, $\vee$, not, we define some operations of constraints manipulation.

We distinguish primitive operations manipulating the type `userConstraint`, complex operations manipulating the schema of satisfaction and the various structures defined above and derived operations. Derived operations can be expressed as a combination of other operations. We do not here consider the phase of constraints visualization. We are particularly interested in manipulating and updating these constraints.

To illustrate the operations, we consider the structures of reference: `satisfactionSchema` reflecting the schema of satisfaction and `constraintList` representing a list of elements of the type `userConstraint`:

`Type constraintList = < userConstraint >.`

We use the conventional signature notation for an operator:

- $\rightarrow$: Return of the operation;
- $\times$: Cartesian product of domains.

Example: `constraintList` $\times$ `Integer` $\rightarrow$ `constraintList`.

This operation admits two parameters of types: `constraintList` and `Integer`. `constraintList` is the result type.

To illustrate the operations with examples, we consider the following operational context:

schema1, schema2:	variables of type `satisfactionSchema`;
LPsecondary:	variable of type `LP`;
stiltedComplexity:	variable of type `complexityLevel`;
list1, list2:	variables of type `constraintList`;

6.2.1 *Primitive operations*

The main primitive operations are the following:

- ***`Insert:`*** `constraintList × userConstraint → constraintList.`

The `Insert` operation adds a constraint to the initial list according to its value of preference, its value of complexity and its relevance. The length of the return list is superior to the initial one.

- ***`Delete:`*** `constraintList × userConstraint → constraintList.`

The `Delete` operation removes a constraint from the initial list. The length of the return list is lower than the initial one. This operation is not defined if the constraint to delete does not belong to the initial list.

- ***`Modify:`*** `userConstraint × userPreference → userConstraint.`

The `Modify` operation updates the value of preference of a constraint. This operation returns a constraint with the same value of complexity and relevance.

- ***`Select:`*** `constraintList × selectionCondition → constraintList.`

The `Select` operation allows extracting a list of constraints verifying a selection criteria σ. The criteria σ can be applied to the parameters: identifier, specification and relevance characterizing the constraints. The parameters preferenceValue and complexityValue are not supported by the Select operation. The selection according to these parameters can be expressed via the Extract operation.
A selection is a triplet: attribute, comparator, value surrounded by quotation marks. This operation returns a list, eventually empty, such as the length is lower or equal to the initial one.

Example: To select the constraints with relevance (related to the application weight parameter) lower than 0.2:
list2 = `Select` (list1, "relevance < 0.2").

6.2.2 *Complex operations*

Primitive operations identified above are not sufficient to manage the level of expression of updating constraints. Often, the intention to change expressed by users may be represented at a level of abstraction higher than the of primitive operations level. Composite operations are more powerful since a user does not need to go through each step of the sequence of primitive operations to achieve the desired operation. Therefore, the operations make constraint handling much easier and more efficient. They are more easily usable and understandable because their intention is explicit, contrary to a sequence of primitive operations with the same result.
We define a subset of the main complex operations, available for the stakeholders, to modify the schema of satisfaction. This subset is: an extraction of constraints (Extract), a deletion of constraints (Cut), an insertion of a list of constraints (Insert), an update of the

preferences on a list of constraints (Modify), an evaluation of the temporal complexity of the constraints (Evaluate) and a respect of a simulation time (Respect).

- ***`Extract:`*** `satisfactionSchema × userPreference → LP.`

The polymorphic `Extract` operation obtains a level of preference LP (corresponding to the specified `userPreference`) from the structure satisfactionSchema.
Example: To extract the secondary constraints (Figure 7):
LPsecondary = `Extract` (schema1, SECONDARY).

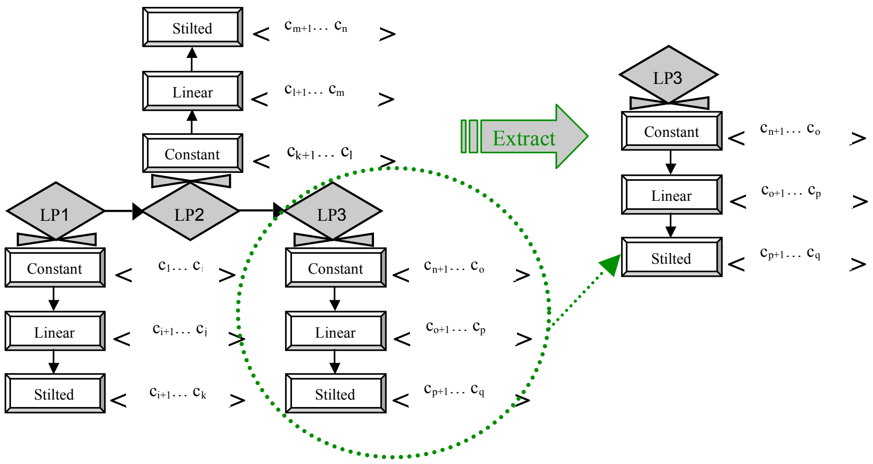


Figure 7. *Illustration of the operation* `Extract`.

- ***`Extract:`*** `LP × conceptualComplexity → complexityLevel.`

The polymorphic `Extract` operation obtains a level of complexity (corresponding to the specified `conceptualComplexity`) relative to a given LP.
Example: To extract the secondary constraints with a stilted complexity:
stiltedComplexity = `Extract` (LPsecondary, STILTED) = `Extract` (`Extract` (schema1, SECONDARY), STILTED).

- ***`Cut:`*** `satisfactionSchema × userPreference → satisfactionSchema.`

The polymorphic `Cut` operation is equivalent to a deletion of a level of preference from a structure satisfactionSchema, i.e., it suppresses the constraints with a value of preference equal to the specified `userPreference`. This operation returns a structure of constraints satisfactionSchema, eventually empty, included in the initial structure.
Example: To cut the secondary constraints (Figure 8):
schema2 = `Cut` (schema1, SECONDARY).

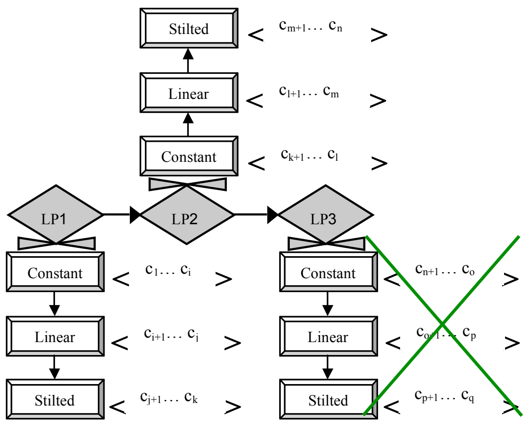


Figure 8. *Illustration of the operation* `Cut`.

Remark. The deletion of several levels of preference is equivalent to a combination of several `Cut` operations.
Example: To cut the secondary and the flexible constraints:
schema2 = `Cut` (`Cut` (schema1, SECONDARY), FLEXIBLE).

- ***`Cut:`*** `LP × conceptualComplexity → LP.`

The polymorphic `Cut` operation allows deleting a level of complexity belonging to a given LP. This operation returns a level of preference LP, eventually empty, included in the initial LP.
Example: To cut the secondary constraints having a stilted complexity:
LPsecondary = `Cut` (`Extract` (schema1, SECONDARY), STILTED).

Remark. The deletion of several levels of complexity is equivalent to a combination of several Cut operations.
Example: To cut the secondary constraints having a stilted or linear complexity:
LPsecondary = `Cut` (`Cut` (`Extract` (schema1, SECONDARY), STILTED), LINEAR).

- ***`Cut:`*** `satisfactionSchema × conceptualComplexity → satisfactionSchema.`

The polymorphic `Cut` operation allows deleting a level of complexity from the structure satisfactionSchema, i.e., to delete the constraints with a value of complexity equal to the specified `conceptualComplexity`. This operation returns a structure satisfactionSchema, eventually empty, included in the initial structure.

Example: To cut all the constraints of stilted complexity:
schema2 = `Cut` (schema1, STILTED).

Remark. The deletion of several levels of complexity is equivalent to a combination of elementary Cut operations.

Example: To cut all the constraints of stilted or linear complexity:
schema2 = `Cut` (`Cut` (schema1, STILTED), LINEAR).

- ***`Cut:`*** `satisfactionSchema × constraintList → satisfactionSchema.`

The polymorphic `Cut` operation deletes a list of constraints from the structure satisfactionSchema. This operation returns a structure satisfactionSchema, eventually empty, included in the initial structure of constraints. It is not defined if the constraints to cut do not belong to the structure satisfactionSchema.
Example: To cut the constraints with relevance less than 0.2 from schema1:
schema2 = `Cut` (schema1, `Select` (schema1, " relevance <0.2 ").

- ***`Insert:`*** `satisfactionSchema × constraintList → satisfactionSchema.`

The `Insert` operation adds a list of stakeholders' constraints to the structure satisfactionSchema. The added constraints will be classified according to their value of preference, their value of complexity and their relevance. This operation returns a structure satisfactionSchema such as its cardinality (of constraints) is greater than the initial one.

- ***`Modify:`*** `constraintList × userPreference → constraintList.`

The `Modify` operation updates the value of preference of a constraints list while preserving the same value of complexity and relevance. It returns a list of constraints with the same length as the initial one.

- ***`Evaluate:`*** `satisfactionSchema → Time.`

The `Evaluate` operation estimates the temporal complexity of the constraints of the structure `satisfactionSchema`. It returns an evaluation of the total time to satisfy all the constraints.

- ***`Respect:`*** `satisfactionSchema × Time → Boolean.`

The `Respect` operation returns a Boolean value relative to the state of satisfaction within the required time.

6.2.3 Derived operations

We define the derived operation `Replace`, allowing the reorganization of the constraints according to a new value of preference:

- ***`Replace:`*** `satisfactionSchema × userPreference × conceptualComplexity × userPreference → satisfactionSchema.`

The `Replace` operation consists in modifying, in a structure satisfactionSchema, the place of the constraints verifying a given `userPreference` and `conceptualComplexity`. This leads us to select, in the initial structure satisfactionSchema, the constraints verifying the specified `userPreference` and the `conceptualComplexity` (operation `Extract`), to update their preferences (operation `Modify`), to insert them in another level of preference according to their new value of preference, while preserving the same values for complexity and application relevance (operation `Insert`) and finally to delete from the structure satisfactionSchema, the constraints selected for modification (operation `Cut`). The

operation `Replace` returns a structure `satisfactionSchema` with the same cardinality (a number of constraints) that the initial structure.
Example: To replace at the level of preference LP3, the constraints of stilted complexity and having a hard value of preference (Figure 9):
schema2 = `Replace` (schema1, HARD, STILTED, SECONDARY).
The expression of the request using the operations `Cut`, `Insert`, `Modify` and `Extract` takes the following form:
schema2 = `Cut` (`Insert` (schema1, `Modify` (`Extract` (`Extract` (schema1, HARD), STILTED), SECONDARY)), `Extract` (`Extract` (schema1, HARD), STILTED)).

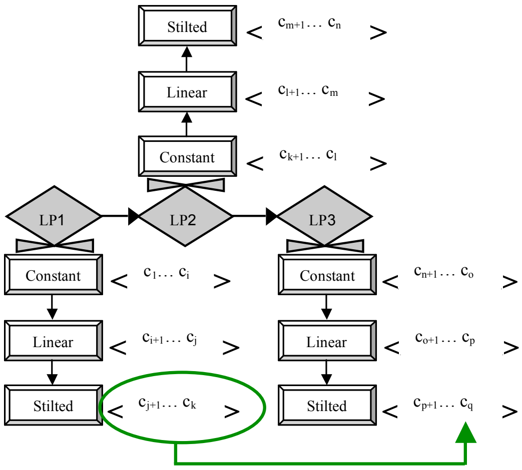


Figure 9. *Illustration of the operation* `Replace`.

Remark. The replacement of a level of complexity belonging to several levels of preference is equivalent to a combination of `Replace` operations.

Example: to replace at the level of preference LP3, the constraints of stilted complexity and of a value of preference hard or flexible, is performed in two times (Figure 10):
schema2 = `Replace` (`Replace` (schema1, HARD, STILTED, SECONDARY), FLEXIBLE, STILTED, SECONDARY).

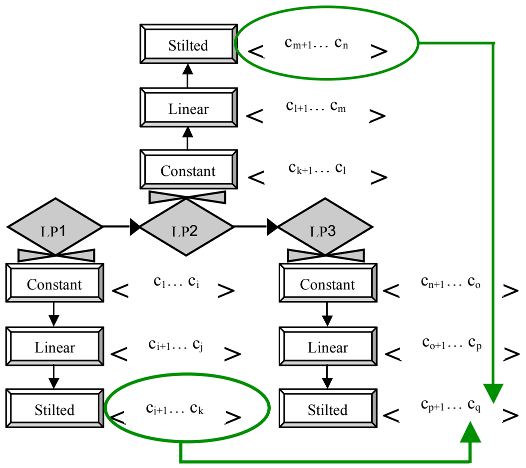


Figure 10. *Illustration of the operation* `Replace`.

The phase of negotiation allows establishing from the schema of satisfaction, the operational graph including the effective constraints to satisfy during the resolution. The operations applied by the stakeholders are controlled by the system in order to visualize the consequences of the modifications (in term of cost) made on the list of constraints. Indeed, the system is based on the complexity of the constraints to negotiate with the stakeholders and to convince them to modify their choices and to propose some disclaimers. The stakeholders' choices determine the analyst's margin and influence the quality of the solution that can be proposed by the system.

7. Experimentation

We present an experimentation of this framework in order to reduce the streaming risk in the watershed of Haute-Durdent (Seine Maritime, France) while satisfying the constraints required by the stakeholders (the farmers in this application).

Streaming is the accumulation of outflows on parcels, due to a weak infiltration of water into the soil. Above a certain rate of rain, a soil doesn't absorb any more water that flows according to the natural slope. This causes losses of land (erosion) and a significant deterioration of the agricultural soil's surface. Important damages can be caused on the habitat and on the plantations, notably in the downstream areas [23].

The watershed of Haute-Durdent (16 km², Pays de Caux) has 450 parcels shared between a dozen of farms. Thirteen types of occupations: beet, wheat, wood, rape, fodder crops, escourgeon, fallow, linen, corn, potato, pea, prairie and village are placed on the parcels. To reduce the streaming risk, the retained strategy consists in re-distributing crops on the parcels of the watershed while taking into account the farmers' constraints. The wood and village are considered as invariable and are not taken into account in the process of crops re-affectation. In this experimentation, a solution to the problem is a distribution of crops on the parcels, which minimizes the streaming risk and maintains the initial threshold of satisfaction for the stakeholders' constraints.

The spatial units in our application (the agricultural parcels) are organized according to a directed graph structure. The spatial relationship between these spatial units are of type: Upstream - downstream (from... to) corresponding to the transfer of water / risk between them (Figure 11).

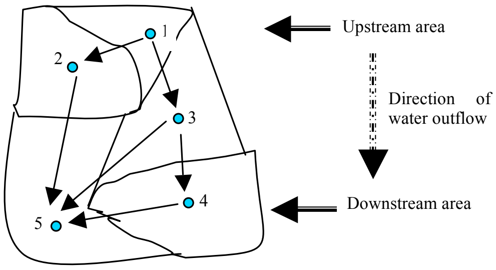


Figure 11. *The spatial relationship and transfer of risk between parcels.*

We present in this part an example of manipulations of farmers' constraints. Several farmers provide their constraints, each in his respective farm. They use the defined operations (`Insert` operation) of constraint manipulation to express the constraints that must be considered by the system. We consider 31 stakeholder constraints: 16 hard (LP1), 9 flexible (LP2) and 6 secondary (LP3) that represent the schema of satisfaction (schema1, Figure 15). The constraints are essentially related to the spatial placement of crops, the agronomic requirements of slopes, types of soil and cycles of rotation as well as the economic objectives of production for crops in each farm.

Although the number of constraints is 31, these constraints have multiple instances. For example, the constraint of adjacency (Figure 14) that requires the absence of pro-runoff crops in neighbouring parcels, must be checked for all parcels (450 parcels). Since we cannot detail all constraints taken into account, we present, in the following, 3 examples of constraints.

Example 1 (Constraint 1): The constraint of slope (Figure 12) requires that the flax in the farm 'Leber' must be placed on parcels with slope < 10:

```
Constraint: Slope
Specification: "Requirement of slope"
Type: Alphanumeric
Relevance: 0.2
Preference: Hard
Entity: Parcel [Occupation, Farm,]
Request: Slope [Parcel.Occupation = 'flax' and Parcel.Farm = 'Leber'] < 10;
```

Slope: `userConstraint`
`identifier`: Slope `specification`: Requirement of slope `preferenceValue`: Hard `complexityValue`: Constant `relevance`: 0.2

Figure 12. *Example of a constraint of slope.*

Example 2 (Constraint 3): The constraint of Corn (Figure 13) requires the absence of Corn (a pro-runoff crop) in sensitive downstream parcels of the territory:

```
Constraint: Corn
Specification: "Spatial placement of Corn"
Type: Spatial
Relevance: 0.1
Preference: Flexible
Entity: Parcel [Occupation, Position, Sensitive,]
Request: Parcel.Occupation [Parcel.Position = Downstream and Parcel.Sensitive = True] <> Corn;
```

Corn: `userConstraint`
`identifier`: Corn `specification`: Spatial placement of corn `preferenceValue`: Flexible `complexityValue`: Linear `relevance`: 0.1

Figure 13. *Example of a constraint of corn.*

Example 3**:** The constraint of adjacency (Figure 14) requires to not placing pro-runoff crops in neighbouring parcels:

```
Constraint: Adjacency
Specification: "Spatial distribution of pro-runoff crops"
Type: Spatial
Relevance: 0.3
Preference: Hard
```

```
Entity: Adjacency [ParcelFrom, ParcelTo], Parcel [Occupation,],
Occupation [Id, Type,]
Request: Not [(Adjacency.ParcelFrom= ParcelX and Adjacency.ParcelTo=
ParcelY) and (ParcelX.Occupation= OccupationX.Id and
OccupationX.Type = Pro-runoff) and (ParcelY.Occupation=
OccupationY.Id and OccupationY.Type = Pro-runoff)];
```

Adjacency: `userConstraint`
`identifier`: Adjacency `specification`: Spatial distribution of crops `preferenceValue`: Hard `complexityValue`: Stilted `relevance`: 0.3

Figure 14. *Example of a constraint of adjacency.*

The phase of negotiation, based on operations of constraints manipulations, allowed deleting all constraints with a stilted complexity and those with a level of preference secondary (LP3). This negotiation can be expressed through two operations of constraints manipulations:

- Operation 1: schema2 = `Cut` (schema1, STILTED).

This operation deletes all constraints with a stilted complexity (Figure 15).

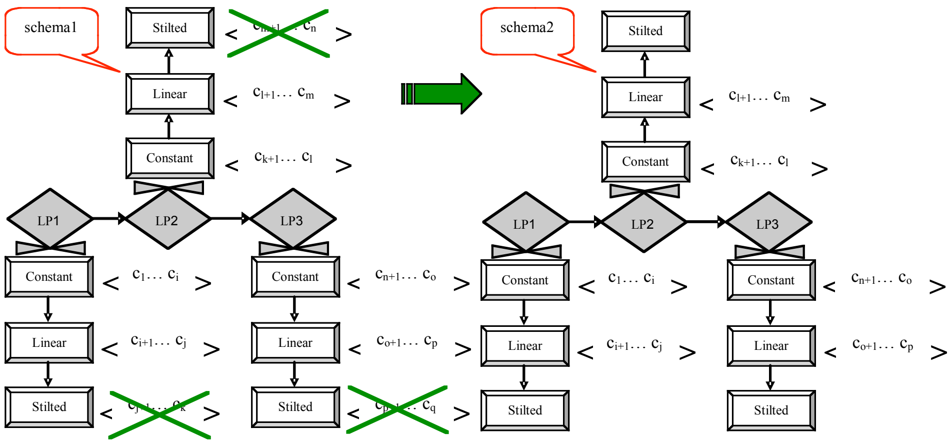


Figure 15. *The result of the operation cut (operation 1). shema1 and schema2 are* `satisfactionSchema` *structures.*

- Operation 2: schema3 = `Cut` (schema2, SECONDARY).
  This operation deletes the constraints in the level of preference secondary (Figure 16).

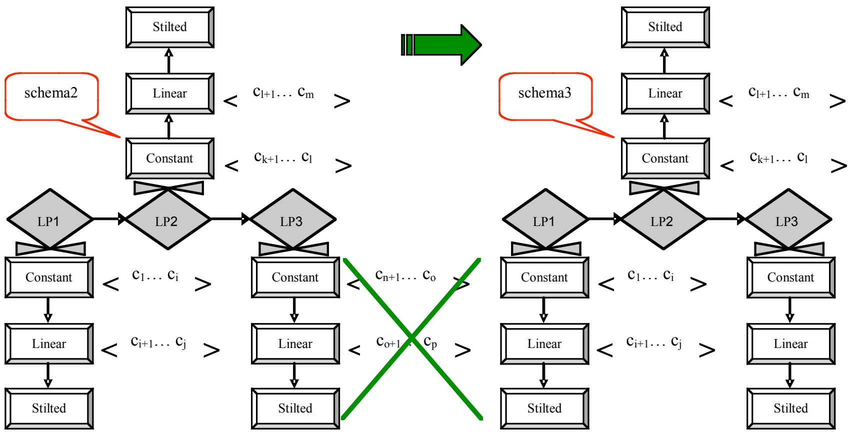


Figure 16. *The result of the operation cut (operation 2). Schema3 is a* `satisfactionSchema` *structure.*

We thus retain 20 constraints (Figure 17) in agreement with decision-makers. These constraints represent the operational graph of satisfaction (schema3).

	Constant	Linear	Stilted	Total
Hard	3	12	1	16
Flexible	2	3	4	9
Secondary	3	2	1	6
Total	8	17	6	31

Negotiation

	Constant	Linear	Stilted	Total
Hard	3	12	0	15
Flexible	2	3	0	5
Secondary	0	0	0	0
Total	5	15	0	20

Figure 17. *Constraints required by stakeholders and retained after the negotiation.*

The negotiation phase is ensured by means of primitive and complex operations of constraints manipulation. For reasons of simplicity, we visualized only 2 operations: SCHEMA2 = Cut (schema1, stilted) and schema3 = Cut (SCHEMA2, SECONDARY) (Figures 15 and 16). These two operations summarize operations applied by stakeholders during the negotiation phase. Indeed, this phase is collaborative, based on a set of Graphic User Interfaces that implement the various operations of constraints manipulations. Through the GUI, the system assists stakeholders in the expression of their constraints and visualizes the consequences of applied operations on the constraints. In this article, the user interfaces are not presented. We essentially focus on the model and the constraints operators.

The used optimization is a hybrid method based on the evolutionary strategy [44] and the simulated annealing [16]. Starting from the real distribution of crops already available, the optimization strategy is based on a permutation of crops between two parcels of territory in ... k. The value of streaming risk is evaluated using a Digital

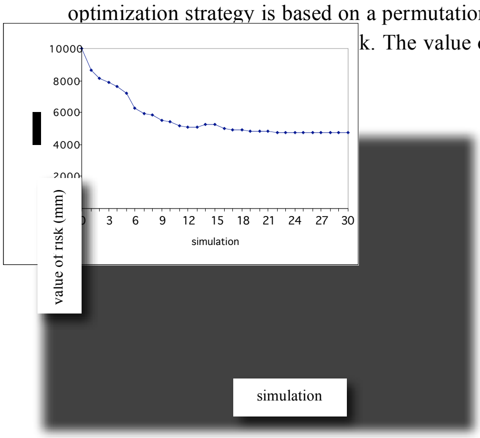


Figure 18. *Value of* streaming *risk during 30 simulations.*

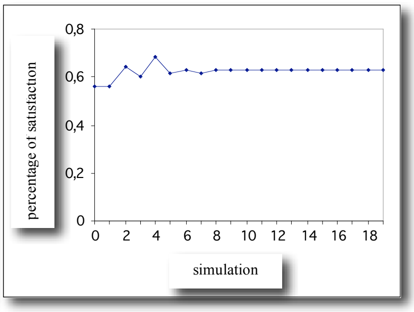


Figure 19. *Average rate of satisfaction for the stakeholders' constraints.*

The results show a reduction of the risk value (Figure 18) while preserving (and improving) the initial rate of satisfaction for the stakeholders' constraints (Figure 19). The system converges after some simulations towards a stable state with any permutation of crops between the parcels.

To verify the conformity of the results in relation to the agronomic and spatial requirements, we visualized on maps, the distribution of crops before and after the optimization process (Figures 20, 21 and 22).

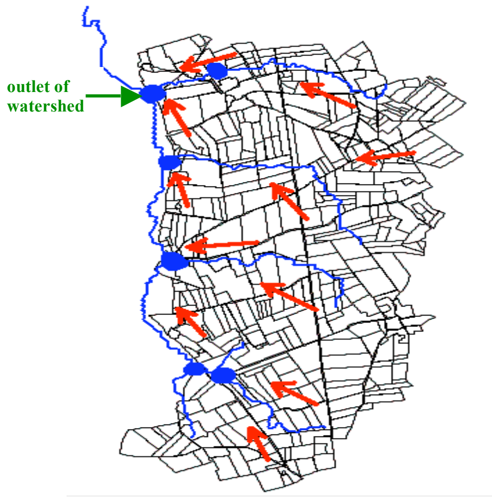


Figure 20. *Flooded zones and direction of drainage through the watershed. Arrows indicate the direction of water flow through the watershed, upstream to downstream towards the outlet. Flooded zones indicate the areas most susceptible to streaming.*

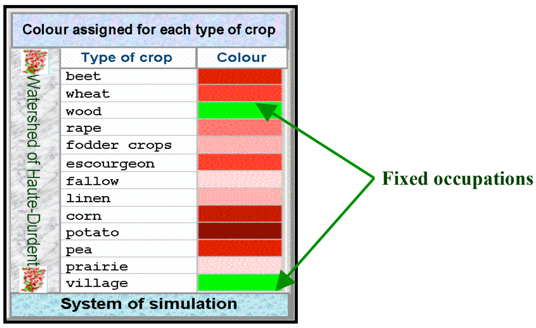


Figure 21. *Colour assigned for each type of crop is darker as the crop is streaming. The fixed occupations are represented in green.*

As shown in Figure 20, the simulation system distributes initially existing big sets of crops causing streaming in order to limit the streaming risk. The proposed distribution map puts anti-streaming crops in accordance to out-flow in the place of biggest streaming crops (Figure 22, circled parts). The analysis of the soil occupation proposed by the system shows that the surfaces of prairie (anti-streaming crop) increased to the detriment of very dripping crops (that does not promote water absorption), notably corn and wheat. This explains the reduction of the risk value.

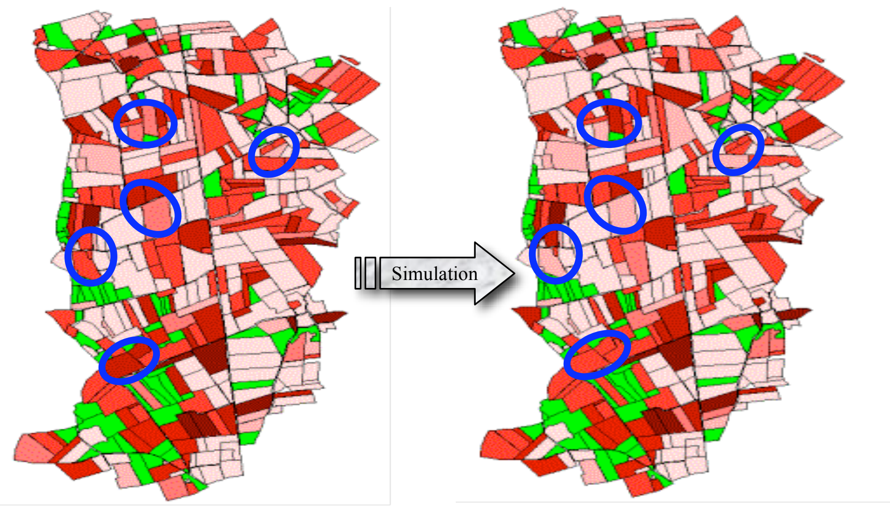


Figure 22. *Distribution of crops before (on the left) and after the optimization.*

8. Conclusion and perspectives

We presented in this article a framework to manage hard and soft constraints in the context of geographic decision support. We concentrated on the formal definition of the problem to categorize constraints according to the different stakeholders' needs. The proposed framework provides algebraic operations to manipulate the constraints to simplify an over-constrained system.

Primitive and complex operations are available to stakeholders to allow them expressing and relaxing their constraints. The complex operations are composite and grouping logical sequences of primitive operations. They incorporate information on their impact on the schema of constraints satisfaction. In addition to their specification, the complexity is also evident in the effects of these operations. If the effects of primitive operations are relatively minor, the cumulative effects of all intermediate operations performing a complex operation are consistent.

While the compositions of complex operations cannot be exhaustively defined (complex operations are infinite as new compositions of operations can always be defined), the question arises: how to provide in practice guidelines to guide their application more optimized and automated as well as to provide a rich and expressive framework for constraints handling.

The manipulation of constraints is based on a functional language of constraints representation. The decision-makers are the actors of the system through the definition and the updating of their constraints. The advantage of the constraints representation language is its flexibility of manipulation at the data-processing level. However, it does not represent the level of interaction adapted to the non-familiar actors with data-processing tool.

The application of the developed system on a problem of risk management shows its contribution in the face of geographic applications. This contribution concerns simultaneously the satisfaction of required constraints and the integration of stakeholders in the resolution process. However, a thorough experimentation must be pursued in order to elaborate a finer valuation of the developed system and its efficiency to solve over-constrained problems with a user-friendly interface. It would also be interesting to experiment the system in the face of others problems involving spatial decision-making and environmental risk management.

References

1. K. Al-Kodmany. "Online tools for public participation", Government InformationQuarterly 18(4): 329-341, 2001.
2. J.M. Attonaty, M.H. Chatelin and F. Garcia. "Interactive simulation modeling in farm decision making", Computers and Electronics in Agriculture, 22(2-3) 157-170, 1999.
3. O. Barreteau and F. Bousquet. "SHADOC: a multi-agent model to tackle viability of irrigated systems", Annals of Operations Research, 94: 139-162, Kluwer Academic Publishers, 2000.
4. O. Barreteau. "The joint use of role-playing games and models regarding negotiation processes: characterization of associations", Journal of Artificial Societies and Social Simulations, 6 (2), 2003. http://jasss.soc.surrey.ac.uk/6/2/3.html.
5. İ. Birbil, G. Bouza, J.B.G. Frenk and G. Still, Equilibrium constrained optimization problems, European Journal of Operational Research, Volume 169, Issue 3, pages 1108-1127, 2006.
6. A. Bockmayr and J.N. Hooker, Constraint Programming, Handbooks in Operations Research and Management Science, Volume 12, pages 559-600, 2005.
7. F. Bousquet, O. Barreteau, P. D'aquino, M. Etienne, S. Boissau, S. Aubert, C. Le Page, D. Babin and J.-C. Castella. "Multiagent systems and role games: an approach for ecosystem co-management", Complexity and Ecosystem Management: The Theory and Practice of Multi-agent Approaches (Ed, Janssen, M.), pp. 248-285, Edward Elgar Publishers, 2002.
8. J.G. Carter, I. White and J. Richards, Sustainability appraisal and flood risk management, Environmental Impact Assessment Review, Volume 29, Issue 1, pages 7-14, 2009.
9. P. Charman. "A constraint-based approach for the generation of floor plans", in Proc. 6th IEEE International Conference Tools with Artificial Intelligence (ICTAI'94), pp. 555-561, New Orleans, USA, 1994.
10. I. Charon and O. Hudry, The noising methods: A generalization of some metaheuristics, European Journal of Operational Research, Volume 135, Issue 1, pages 86-101, 2001.
11. F. Chen, E.T. Kanemasu, L.T. West and F. Rachidi. "Analysis of land use and simulation of soil erosion with GIS for the semi-arid region of Morocco", Géo-observateur, pp. 55-75, 1999.
12. B. De Marchi, S. Funtowicz, S. Lo Cascio and G. Munda. "Combining participative and institutional approaches with multi-criteria evaluation: An empirical study for water issues in Troina, Sicily", Ecological Economics, 34: 267-282, 2000.
13. A. Dury, F. Leber and V. Chevrier. "A reactive approach for solving constraint satisfaction problems: assigning land use to farming territories", Artificial Intelligence, Intelligent Agents V, 1555: 397-412, Springer, 1999.
14. J.R. Dymond, H.D. Betts and C.S. Schierlitz, An erosion model for evaluating regional land-use scenarios Environmental Modelling & Software, Volume 25, Issue 3, pages 289-298, 2010.
15. S. Effati and H. Roohparvar, Iterative dynamic programming for solving linear and nonlinear differential equations, Applied Mathematics and Computation, Volume 175, Issue 1, pages 247-257, 2006.
16. R. Faber, T. Jockenhövel and G. Tsatsaronis, Dynamic optimization with simulated annealing, Computers & Chemical Engineering, Volume 29, Issue 2, pages 273-290, 2005.
17. J. Ferber. "Eco problem solving: How to solve a problem by interactions", in Proc. 9th Workshop on Distributed Artificial Intelligence, pp. 113-128, USA, 1989.
18. T. Gajdos, J.-M. Tallon and J.-C. Vergnaud, Representation and aggregation of preferences under uncertainty, Journal of Economic Theory, Volume 141, Issue 1, pages 68-99, 2008.
19. M. Ghallab, D. Nau and P. Traverso, Constraint Satisfaction Techniques, Automated Planning, pages 167-191, 2004.

20. J.S. Ha and P.H. Seong, A human–machine interface evaluation method: A difficulty evaluation method in information searching (DEMIS), Reliability Engineering & System Safety, Volume 94, Issue 10, pages 1557-1567, 2009.
21. K. Hammouche, M. Diaf and P. Siarry, A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem, Engineering Applications of Artificial Intelligence, 2009.
22. O. Hasançebi, Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures, Computers & Structures, Volume 86, Issues 1-2, pages 119-132, 2008.
23. W. Jaziri and T. Paquet, A Multi-Agent Model and Tabu search Optimization to Manage Agricultural Territories, GeoInformatica- An International Journal on Advances of Computer Science for GIS, Volume 10, Issue 3, pages 337-357, Springer, Kluwer Academic Publishers, USA, 2006.
24. W. Jaziri, Multi-Scale Optimization for the Management of Run-off Risks in Agricultural Watersheds, IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews, Volume 37, Issue 4, pages 573-582, 2007.
25. V. Jetten, J. Boiffin and A. De Roo. "Defining monitoring strategies for runoff and erosion studies in agricultural catchments: a simulation approach", Soil Science, 47: 579-592, Lippincott Williams & Wilkins, 1996.
26. R. Kingston, S. Carver, A. Evans and I. Turton. "Web-based public participation geographical information systems: an aid to local environmental decision-making", Computers, Environment and Urban Systems, 24(2): 109-125, 2000.
27. F. Le Ber, V. Chevrier and A. Dury. "A multi-agent system for the simulation of land use organization", in Proc. 3rd IFAC/CIGR Workshop on Artificial Intelligence in Agriculture, pp. 182-187, Japan, 1998.
28. F. Le Ber and M. Benoît. "Modelling the spatial organization of land use in a farming territory, example of a village in the plateau lorrain", Agronomie: Agriculture and Environment, 18: 101-113, EDP Sciences, 1998.
29. L. Leenen, T. Meyer and A. Ghose, Relaxations of semiring constraint satisfaction problems, Information Processing Letters, Volume 103, Issue 5, pages 177-182, 2007.
30. L.Y. Liang and W.C. Chao, The strategies of tabu search technique for facility layout optimization, Automation in Construction, Volume 17, Issue 6, pages 657-669, 2008.
31. N. Lind, M. Pandey and J. Nathwani, Assessing and affording the control of flood risk, Structural Safety, Volume 31, Issue 2, pages 143-147, 2009.
32. M.K. Luhandjula, Fuzzy stochastic linear programming: Survey and future research directions, European Journal of Operational Research, Volume 174, Issue 3, pages 1353-1367, 2006.
33. A. MacFarlane and A. Tuson, Local search: A guide for the information retrieval practitioner, Information Processing & Management, Volume 45, Issue 1, pages 159-174, 2009.
34. R. Marinescu, R. Dechter, AND/OR Branch-and-Bound search for combinatorial optimization in graphical models, Artificial Intelligence, Volume 173, Issues 16-17, pages 1457-1491, 2009.
35. P. Martin. "Reducing flood risk from sediment-laden agricultural runoff using intercrop management techniques in northern France", Soil and Tillage Research, 52: 233-245, Elsevier, 1999.
36. P. Martin, F. Papy and A. Capillon. "Agricultural field state and runoff risk: proposal of a simple relation for the silty-loam-soil context of the Pays de Caux (France)", in 10th International Soil Conservation Organization Meeting, pp. 293-299, USA, 2001.
37. F.A. Mussa-Ivaldi and Z. Danziger, The remapping of space in motor learning and human–machine interfaces, Journal of Physiology-Paris, Volume 103, Issues 3-5, pages 263-275, 2009.
38. H. Posthumus, C.J.M. Hewett, J. Morris and P.F. Quinn, Agricultural land use and flood risk management: Engaging with stakeholders in North Yorkshire, Agricultural Water Management, Volume 95, Issue 7, pages 787-798, 2008.
39. B. Roy, Multicriteria methodology for decision aiding, Kluwer Academic Publishers, Dordrecht, Netherlands, 1996.
40. T. Schiex. "Possibilistic constraint satisfaction problems or How to handle soft constraints? ", in Proc. 8th Int. Conf. on Uncertainty in Artificial Intelligence (UAI-92), pp. 268-275, Stanford, CA, USA, 1992.
41. R.M. Soland. "Multicriteria optimization: a general characterization of efficient solutions", Decision Sciences, 10(1): 26-38, Blackwell Publishing, 1979.
42. A. Tsoukiàs, From decision theory to decision aiding methodology, European Journal of Operational Research, Volume 187, Issue 1, pages 138-161, 2008.
43. P. Van Beek, Backtracking search algorithms, Foundations of Artificial Intelligence, Volume 2, pages 85-134, 2006.
44. Z. Yang, K. Tang and X. Yao, Large scale evolutionary optimization using cooperative coevolution, Information Sciences, Volume 178, Issue 15, pages 2985-2999, 2008.
45. P. Zhou, O. Luukkanen, T. Tokola and J. Nieminen, Effect of vegetation cover on soil erosion in a mountainous watershed, CATENA, Volume 75, Issue 3, pages 319-325, 2008.

Dr. Wassim Jaziri received a Ph.D. in Computer Sciences in 2004 at the Institut National des Sciences Appliquées (INSA) of Rouen. In October 2004, he joined the French Agricultural Research Centre for International Development -CIRAD- Montpellier (France), as a post-doctorate researcher. Currently, he is an assistant professor in computer science in the Higher Institute of informatics and Multimedia of Sfax- Tunisia. He is a member of the Miracl Laboratory (Multimedia InfoRmation systems & Advanced Computing Laboratory**)**. His main interests are Optimization, Constraints satisfaction, Geographic Information Systems, spatio-temporal databases and Ontology.

Prof. Michel Mainguenaud was the dean for education at the Institut National des Sciences Appliquées (INSA) of Rouen. He was the head of the national SIGMA-Cassini research group (a CNRS recognized national group to promote research in Geographical Information Systems in France). His main interests are Geographical Information Systems, multi-media database models and query languages: documents, images and maps.