Traitement de problèmes de décision sous incertitude par des problèmes de satisfaction de contraintes.

Hélène Fargier*, **Jérôme Lang***
Roger Martin-Clouaire** , **Thomas Schiex****

* *IRIT, Université Paul Sabatier, 118 Route de Narbonne*
31062 Toulouse Cedex
** *INRA/SBIA, Chemin de Borde-Rouge, Auzeville, BP 27*
31326 Castanet-Tolosan Cedex

RÉSUMÉ. *L'approche CSP (problèmes de satisfaction de contraintes) offre un langage de représentation et surtout des méthodes de calcul pour des problèmes de décision simples, sans incertitude. Cet article est consacré à deux extensions du cadre des CSP permettant de traiter des problèmes de décisions sous incertitude. Ces extensions reposent sur une distinction entre deux types de variables : les variables de décision, qui sont contrôlables par l'agent, et les variables contingentes, qui sont incontrôlables. Les valeurs que prennent ces dernières dépendent de l'occurrence d'événements incertains, au sujet desquelles on dispose d'une connaissance soit qualitative soit probabiliste — ce qui nous donnera deux extensions différentes, respectivement les CSP mixtes et les CSP probabilistes. Dans le cas probabiliste, nous donnons deux algorithmes pour calculer respectivement des décisions résolvant le problème avec une probabilité maximale, et des décisions conditionnelles associant une décision adéquate à chaque situation pour laquelle c'est possible.*

ABSTRACT. *The Constraint Satisfaction approach (CSP) offers a representation framework and algorithmic tools for solving simple decision problems, without uncertainty. This paper presents two extensions of the CSP framework aimed at solving decision problems under uncertainty. These extensions rely on a partition of the variables: the decision variables, which are under the control of the agent and the contingent variables, which are uncontrollable; the values of the latter depend on the occurrence of some uncertain events which are supposed to be characterized either qualitatively or probabilistically. Depending on the case, this will lead to two different extensions respectively called mixed CSP and probabilistic CSP. In the probabilistic case, we give two algorithms that compute respectively decisions that solve the problem with maximum probability and conditional decisions that aim at providing an adequate solution for each situation.*

MOTS-CLÉS *: Satisfaction de contraintes, aide à la décision.*
KEY WORDS *: Constraint satisfaction, decision support.*

1 Introduction

Résoudre un problème de décision, c'est généralement choisir entre plusieurs alternatives, exprimées le plus souvent de façon implicite ; ce qui revient à trouver les décisions qui sont les plus appropriées par rapport à la spécification du problème. On peut identifier différents types de problèmes à partir des deux caractéristiques suivantes :

- les décisions sont prises à un instant précis unique (bien qu'elles puissent avoir une structure temporelle, par exemple former un plan) ou sont élaborées en une séquence d'étapes, chacune tenant compte d'informations provenant des précédentes ;
- les spécifications (contraintes, critères) qui définissent implicitement les alternatives sont sujettes ou non à incertitude. Elles sont incertaines si elles dépendent de circonstances dont l'occurrence est incomplètement connue (incertaine).

La prise en compte de décisions séquentielles, comme l'existence de spécifications incertaines, augmentent la complexité du problème de décision [1]. Cet article, consacré au cas des décisions non séquentielles sous spécifications incertaines, propose une approche du problème par satisfaction de contraintes. Le choix du cadre des CSP est motivé par la possibilité de réutiliser des outils calculatoires efficaces.

Jusqu'à présent, les problèmes de décision pris en compte par les approches CSP (Problèmes de Satisfaction de Contraintes) ne concernent que le cas le plus simple : décisions non séquentielles, sans prise en compte de l'incertitude. Si l'on veut traiter de problèmes plus complexes, il est essentiel de distinguer deux types de variables : les variables de décision et les variables « contingentes ». Les variables contingentes sont incontrôlables, c'est-à-dire que les valeurs qu'elles prennent dépendent de l'occurrence d'événements qui ne peuvent pas être contrôlés, ni même influencés, par le décideur (que nous appellerons également « agent »). La valeur d'une variable contingente est imposée par le monde extérieur ; l'agent peut n'en avoir qu'une connaissance partielle. Par opposition, c'est l'agent qui fixe les valeurs des variables de décision. Par exemple, le temps (état de l'atmosphère) est une variable contingente tandis que le port d'un parapluie par l'agent est une variable de décision (dont la valeur dépendra du temps qu'il fera). Ne faire aucune différence entre variables contingentes et variables de décision conduirait à des résultats aberrants ; en effet, la valeur d'une variable contingente peut limiter l'ensemble des valeurs acceptables (satisfaisantes) des variables de décision, alors que l'inverse est physiquement impossible. Il est donc nécessaire de définir une approche CSP et des outils de résolution qui permette de tenir compte de cette dichotomie de l'ensemble des variables.

Cet article traite de deux extensions du modèle CSP mettant en jeu à la fois des variables contingentes et des variables de décision : dans un *CSP mixte,* l'incertitude est exprimée de façon qualitative, alors que dans les *CSP probabilistes*, l'incertitude

1. Une troisième source de complexité, que nous n'aborderons pas ici, concerne la prise en compte de la *flexibilité* des contraintes, où la dichotomie classique entre contraintes satisfaisantes et contraintes violées est remplacée par une notion de satisfaction graduelle (voir [**?**, FRE 92, SCH 95, SCH re, BIS 95]).

sur les valeurs des variables contingentes est représentée par une distribution de probabilité. Un cadre formel de représentation est défini, ainsi que des algorithmes qui s'appuient sur une hypothèse d'indépendance des variables contingentes.

Nous considérons successivement deux situations, selon que l'état des connaissances de l'agent sur les valeurs des variables contingentes (l'état du monde) peut ou non évoluer entre la description du problème (et sa résolution) et le moment où la décision doit être prise :

- **API** (« agent partiellement informé ») : l'agent n'apprendra rien de nouveau sur l'état du monde avant de prendre sa décision ; tout ce qu'il peut savoir est déjà représenté dans la spécification initiale du problème. Dans ce cas, nous proposons un algorithme qui fournit une véritable décision, inconditionnelle : celle qui a le plus de chance d'être adéquate.
- **ACI** (« agent complètement informé ») : l'état du monde sera complètement connu avant la prise de décision (éventuellement, *juste avant*). Il est donc utile de calculer « off-line » une *décision conditionnelle*, prête à l'emploi, que l'agent sera en mesure d'adapter en temps réel, dès qu'il connaîtra le monde réel. Dans cette optique nous avons développé un algorithme « anytime » qui fournit un ensemble de décisions et leurs conditions d'applicabilité, ainsi que les probabilités d'occurrence de ces conditions.

Le cas intermédiaire, où des connaissances partielles sur les mondes peuvent être apprises, n'est pas considéré dans cet article.

En Section 2 nous définissons les notions de CSP mixtes et de CSP probabilistes, ainsi que plusieurs types de « solutions ». Nous montrerons que selon qu'on se place dans l'une ou l'autre des situations précédentes (ACI ou API), ce que l'on attend de la résolution du CSP mixte/probabiliste diffère. Puis nous proposerons pour chacune de ces situations un algorithme de calcul d'une décision optimale (Sections 3 et 4), en nous limitant au cas probabiliste (pour le cas mixte, voir [**?**]).

2 CSP mixtes et probabilistes

2.1 Définitions et notations préliminaires

Un *problème de satisfaction de contraintes* classique est un triplet $P = \langle X, D, \mathcal{C} \rangle$, où X est un ensemble de *variables*, chacune prenant ses valeurs possibles dans un domaine (que nous supposerons fini) D_i (avec $D = \times_i D_i$), et $\mathcal{C}$ est un ensemble de *contraintes*. Chaque contrainte C_i de $\mathcal{C}$ lie un ensemble de variables noté $V(C_i)$ et est définie par un sous-ensemble du produit cartésien du domaine de ces variables. Ce sous-ensemble, noté C_i comme la contrainte elle-même, décrit l'ensemble des affectations possibles des variables de $V(C_i)$: la contrainte C_i est satisfaite par une affectation de ces variables si et seulement si cette affectation appartient à l'ensemble des n-uplets admissibles de C_i. Une solution d'un CSP est une affectation des variables qui satisfait toutes les contraintes. L'ensemble de toutes les solutions d'un CSP P est noté $Sol(P)$.

Dans la suite de cet article, nous considérerons quelquefois des affectations partielles, c'est-à-dire des affectations de sous-ensembles Y de X (donc des n-uplets, éléments du produit cartésien des domaines des variables de Y). La concaténation de deux affectations a_1 et a_2 de V_1 et V_2 (où $V_1 \cap V_2 = \varnothing$) est notée (a_1, a_2). C'est une affectation de $V_1 \cup V_2$. La projection d'une affectation a d'un ensemble de variables Y sur un ensemble de variables $Z \subseteq Y$, notée $a_{\downarrow Z}$ est simplement le n-uplet des valeurs des variables de Z dans a. Cette notion et cette notation peuvent être étendues aux ensembles de n-uplets : la projection $R_{\downarrow Z}$ sur Z d'un sous-ensemble R du produit cartésien des domaines des variables de Y est l'ensemble des projections sur Z de tous les n-uplets de R.

2.2 Problèmes de raisonnement, problèmes de décision

La distinction entre variables contingentes et variables de décision est leur contrôlabilité par l'agent ; les variables de décision sont contrôlables, les variables contingentes ne le sont pas. La représentation d'un problème de décision sous incertitude non dégénéré fera appel à ces deux types de variables (c'est l'objet des sous-sections suivantes). Toutefois il est intéressant (et didactique) de commencer par considérer les deux cas dégénérés où seul un des deux types de variables apparaît ; dans ces deux cas, les problèmes peuvent être codés et résolus par des CSP classiques, mais l'interprétation des résultats obtenus dépend de la nature des variables concernées.

Quand toutes les variables du CSP sont des variables de décision (contrôlables), on a affaire à un CSP « purement orienté décision ». C'est l'interprétation la plus courante (bien qu'implicite) dans la littérature sur les CSP (par exemple, les problèmes de conception ou d'ordonnancement...) . Les contraintes restreignent les choix possibles de l'agent pour les valeurs des variables de décision. Toute solution du CSP sera un choix satisfaisant que l'agent pourra appliquer (et constituera donc une décision acceptable).

Quand toutes les variables du CSP sont des variables contingentes (incontrôlables), les contraintes représentent alors des *connaissances* sur les valeurs réelles des variables contingentes. La résolution du CSP ne consiste alors pas nécessairement à trouver une affectation consistante, mais plutôt à trouver la *véritable valeur* de chaque variable *dans le monde réel*. Ce qui signifie que le processus de résolution doit s'interpréter non pas en termes de décision mais de *raisonnement*, comme dans la plupart des approches logiques pour la représentation des connaissances, la différence étant justement dans le formalisme de représentation des connaissances utilisé (ici, le modèle CSP au lieu d'un modèle logique). On a alors affaire à un CSP « purement orienté raisonnement ». Cette interprétation est plus rare dans la littérature, mais pas inexistante. C'est par exemple le cas pour les problèmes d'étiquetage d'arêtes (en vision) de Waltz ou différents problèmes de reconstruction de données à partir d'informations indirectes (voir [**?**] pour des exemples liés à la biologie moléculaire). On peut enfin citer les puzzles logiques comme « trouver qui habite dans quelle maison et boit quelle boisson ... ».

La différence fondamentale entre ces deux types de CSP repose donc sur l'interprétation qu'il convient de faire des données (variables et contraintes) et des résultats donnés par le mécanisme de résolution. En particulier, il est intéressant d'étudier ce qui signifie l'existence d'une solution unique, de plusieurs solutions ou de l'absence de solution pour un CSP purement orienté décision et pour un CSP purement orienté raisonnement.

- Le CSP a une *solution unique*. C'est le cas le plus simple. S'il s'agit d'un problème de décision, cette solution est l'unique affectation possible des variables de décision. Si le CSP est purement orienté raisonnement, l'unique solution fournit la valeur réelle des variables contingentes (pourvu que la connaissance représentée par les contraintes soit correcte) ; la situation est alors particulièrement favorable puisqu'aucune incertitude ne demeure sur la valeur réelle des variables.
- Le CSP a *plusieurs solutions*. Dans le cas d'un problème de décision, la situation est plutôt favorable car une quelconque des solutions devrait satisfaire l'agent. De plus, si le CSP lui donne toutes les solutions, il pourra choisir n'importe laquelle d'entre elles (ou ajouter de nouveaux critères ou de nouvelles contraintes pour discriminer davantage). À l'inverse, si le CSP est purement orienté raisonnement, l'existence de plusieurs solutions signifie qu'il n'a pas été possible de trouver avec certitude la valeur réelle des variables. Donner une seule solution reviendrait à sélectionner arbitrairement une situation qui apparaît possible compte tenu des contraintes explicitées dans le CSP mais qui n'est pas nécessairement la situation réelle ; donner l'ensemble des solutions est plus approprié, bien que cela puisse induire un coût plus important. On pourrait donc dire que plus le problème a de solutions, plus c'est « confortable » si on a un problème de décision et plus c'est « inconfortable » si on a un problème de raisonnement.
- Le CSP *n'a pas de solution* (c'est-à-dire qu'il est inconsistant). Si le CSP est purement orienté décision, cela signifie que le problème est surcontraint (l'utilisateur a probablement donné trop de contraintes, ou des contraintes trop dures). Si le CSP est purement orienté raisonnement, le diagnostic est totalement différent : une contrainte au moins n'a pas été spécifiée correctement (car les variables ont chacune une valeur dans le monde réel). Dans ce cas, il existe une incompatibilité entre le monde réel et certaines contraintes-connaissances.

Enfin, notons que les requêtes adressées aux deux types de problèmes semblent sensiblement différentes. Dans un problème de décision, la principale question est de déterminer si le problème est consistant, et de lui trouver une solution. Même s'il est intéressant, dans le cadre d'un problème de raisonnement, de déterminer la consistance du CSP (ce qui signifie vérifier que les connaissances ne sont pas contradictoires), les requêtes dans ce type de problème concernent plutôt la preuve d'une propriété (comme c'est le cas dans les approches logiques de représentation des connaissances). En termes de CSP, cette dernière requête doit pouvoir être assimilée à un type particulier d'inférence de contraintes.

Plus généralement, les problèmes de décision sous incertitude non dégénérés impliquent les deux types de variables. Nous parlerons alors de problèmes mixtes (décision / raisonnement), utilisant le terme de contrainte mixte pour désigner une contrainte faisant dépendre les valeurs acceptables des variables de décisions des valeurs effectives des variables contingentes. Quand l'incertitude sur les valeurs effectives des variables contingentes est représentée qualitativement, par un ensemble de contraintes-connaissances, on parlera de *CSP mixtes* ; quand elle sera représentée plus quantitativement par une distribution de probabilité, on parlera de *CSP probabilistes*. Comme ces deux extensions du modèle CSP sont assez similaires, nous les présentons ensemble. Les algorithmes que nous donnerons sont adaptés aux CSP probabilistes, mais il existe des algorithmes similaires pour les CSP mixtes [**?**].

2.3 CSP mixtes et CSP probabilistes : définitions

Un CSP mixte est un CSP qui comporte non pas un seul, mais deux ensembles de variables : les variables de décision et les variables contingentes.

Définition 1 *Un* CSP mixte *est un sextuplet* $\mathcal{P} = \langle \Lambda, W, X, D, \mathcal{C}, \mathcal{K} \rangle$ *où :*

- $\Lambda = \{\lambda_1, \ldots, \lambda_p\}$ *est un ensemble des variables contingentes ;*
- $W = W_1 \times \cdots \times W_p$*, où* W_i *est le domaine de* λ_i *;*
- $X = \{x_1, \ldots, x_n\}$ *est un ensemble de variables de décision ;*
- $D = D_1 \times \cdots \times D_n$*, où* D_i *est le domaine de* x_i *;*
- $\mathcal{C}$ *est un ensemble de contraintes, chacune mettant en jeu au moins une variable de décision.*
- K *est un ensemble de contraintes ne portant que sur des variables contingentes.*

Les contraintes sont définies de la même façon que dans les CSP classiques, par des relations. Nous noterons $X(C_i)$ *(respectivement,* $\Lambda(C_i)$*) l'ensemble des variables de décision (respectivement, contingentes) de* C_i*, prise dans* $\mathcal{C}$ *ou* K*.*

Un CSP mixte est donc un CSP muni d'une partition entre les variables de décision et les variables contingentes. Une instanciation de toutes les variables contingentes, généralement notée ω, sera appelée un *monde*. Une instanciation de toutes les variables de décision, généralement notée d, sera appelée une *décision*. Un monde partiel (respectivement une décision partielle) est une instanciation d'un sous-ensemble des variables contingentes (respectivement, des variables de décision).

La distinction importante qui est faite entre les contraintes de K et celles de $\mathcal{C}$ est dûe à leurs natures bien différentes : les contraintes de $\mathcal{C}$ sont des contraintes de décision (elles restreignent les valeurs possibles des variables de décision), éventuellement conditionnées par les valeurs de certaines variables contingentes. Dans ce dernier cas, on parlera de *contraintes de décision mixtes*. L'ensemble des contraintes de décision pures (ne dépendant d'aucune variable contingente) sera noté $\mathcal{C}^*$.

Les contraintes de K, elles, ne portent que sur des variables incontrôlables et représentent des connaissances. Le CSP $\langle \Lambda, W, K \rangle$ est un CSP classique, orienté raisonnement. Une solution de ce CSP est un *monde possible*.

Définition 2 (mondes possibles) *L'ensemble des solutions du CSP* $\langle \Lambda, W, K \rangle$*, désigne l'ensemble des mondes possibles. Il est noté* $Poss(\mathcal{P})$*.*

Cette notion de CSP mixte suppose donc que les connaissances disponibles sur le monde, pour incertaines qu'elles soient, restent purement qualitatives. Dans certains cas, elles peuvent être plus quantitatives, par exemple être de nature statistique ; on les exprimera alors non plus par un ensemble K de contraintes, mais par une distribution de probabilité :

Définition 3 *Un* CSP probabiliste *est un sextuplet* $\mathcal{P} = \langle \Lambda, W, X, D, \mathcal{C}, \surd\nabla \rangle$ *où :*

- $\Lambda = \{\lambda_1, \ldots, \lambda_p\}$ *est un ensemble des variables contingentes ;*
- $W = W_1 \times \cdots \times W_p$*, où* W_i *est le domaine de* λ_i *;*
- $X = \{x_1, \ldots, x_n\}$ *est un ensemble de variables de décision ;*
- $D = D_1 \times \cdots \times D_n$*, où* D_i *est le domaine de* x_i *;*
- $\mathcal{C}$ *est un ensemble de contraintes, chacune mettant en jeu au moins une variable de décision ;*
- $pr : W \to [0, 1]$ *est une distribution de probabilité sur les affectations des variables contingentes.*

Nous n'avons pas encore précisé comment la distribution de probabilité sur les mondes est spécifiée. Il n'est en général pas raisonnable de supposer que *pr* est spécifiée *explicitement*. Généralement, pr sera donnée de manière plus concise. Dans le cas le plus simple, les variables contingentes sont mutuellement indépendantes (*pr* est donnée par des distributions de probabilité individuelles pr_{λ_i} associées à chaque λ_i). On pourrait considérer un cas plus complexe où *pr* est définie par un réseau bayésien : le calcul de *pr* est alors effectué par propagation dans le réseau. En d'autres termes, *pr* est en général donné de manière *implicite* plutôt qu'explicite, et son obtention peut demander un certain effort de calcul. Dans la suite, lorsque nous parlerons de CSP probabilistes, nous ignorerons cette étape de calcul en supposant immédiate l'obtention de pr. Par conséquent, $Poss(\mathcal{P})$, l'ensemble des mondes possibles n'est plus obtenu, comme dans les CSP mixtes [**?**], par la résolution d'un CSP orienté raisonnement, mais défini à partir de *pr* :

Définition 4 (mondes possibles) *Dans un CSP probabiliste, un monde* ω *de W tel que* $pr(\omega) > 0$ *est un monde possible. L'ensemble des mondes possibles est toujours noté* $Poss(\mathcal{P})$.

Les définitions qui suivent (réduction d'une contrainte par une affectation, CSP candidats, mondes couverts par une décision, mondes sympathiques et antipathiques) sont communes aux CSP mixtes et aux CSP probabilistes.

Soit a une affectation d'un sous-ensemble des variables (ici, la notion de *variable* couvre à la fois les variables contingentes et les variables de décision). La *réduction* d'une contrainte C par une affectation a est l'ensemble des affectations des variables non instanciées de C qui sont compatibles avec a selon C :

Définition 5 (Réduction d'une contrainte) *Soit C une contrainte de $\mathcal{C}$ mettant en jeu un ensemble $V(C) = \Lambda(C) \cup X(C)$ de variables. La réduction de C par une affectation de a de V' est la contrainte $Reduce(C, a)$ sur $V(C) - (V' \cap V(C))$ définie par les n-uplets $\{b \in D_{\downarrow V(C)-(V'\cap V(C))} \mid (b, a)_{\downarrow V(C)} \in C\}$. On notera $\mathcal{C}[a] = \{Reduce(C, a), C \in \mathcal{C}\}$ la réduction de $\mathcal{C}$ par a.*

Nous allons utiliser cette notion de réduction lorsque a est un monde ω ou une décision d.

La réduction de $\mathcal{C}$ par un monde ω est en effet un ensemble de contraintes de décision classiques, ne liant que des variables contrôlables, dont les solutions sont les décisions qui sont satisfaisantes s'il s'avère que le monde réel est effectivement ω. En fait, on peut associer à chaque monde ω un problème de décision classique (sans incertitude) $\langle X, D, \mathcal{C}[\omega]\rangle$. Chacun de ces CSP, orienté décision, est un *candidat*. L'un d'eux (et un seul d'entre eux !) représente le problème de décision qu'il suffirait de résoudre si l'on savait exactement quel était le monde réel : le CSP $\langle X, D, \mathcal{C}[\omega^*]\rangle$, correspondant au monde réel ω^*, sera appelé *problème réel*. Mais, puisque l'agent ne sait pas lequel des mondes possibles est le monde réel, il ne connaît pas non plus le problème réel.

Définition 6 (problèmes candidats) *L'ensemble des problèmes candidats induits par $\mathcal{P}$ est défini par $Candidats(\mathcal{P}) = \{\langle X, D, \mathcal{C}[\omega]\rangle \mid \omega \in Poss(\mathcal{P})\}$.*

Parmi les CSP de $Candidats(\mathcal{P})$, certains peuvent être inconsistants : dans ce cas, il est possible que le problème réel n'ait pas de solution.

Réciproquement, la réduction de $\mathcal{C}$ par une décision d définit un CSP $\langle \Lambda, W, \mathcal{C}[d]\rangle$ qui ne met en jeu que des variables contingentes. Les solutions de ce problème sont les mondes pour lesquels d est satisfaisante : on dira que ces mondes sont *couverts* par d. Remarquons que, de façon évidente, ω est une solution de $\langle X, D, \mathcal{C}[d]\rangle$ si et seulement si d est une solution de $\langle X, D, \mathcal{C}[\omega]\rangle$. Ce qui est équivalent à dire que (ω, d) est une solution de $\langle \Lambda \cup X, W \times D, \mathcal{C}\rangle$.

Parmi les mondes possibles, ceux qui sont couverts par au moins une décision, ou, ce qui revient au même, ceux qui induisent un problème candidat consistant, sont des *mondes sympathiques* — les autres sont appelés *antipathiques*. L'ensemble des mondes sympathiques (respectivement antipathiques) pour $\mathcal{P}$ sera noté $Good(\mathcal{P})$ (respectivement $Bad(\mathcal{P})$).

Définition 7 $Good(\mathcal{P}) = \{\omega \in Poss(\mathcal{P}) \mid \exists d \in D, (\omega, d)$ *satisfait* $\mathcal{C})\}$
$Bad(\mathcal{P}) = Poss(\mathcal{P}) \setminus Good(\mathcal{P})$

Si $Bad(\mathcal{P})$ [2] n'est pas vide, la situation risque d'être inconfortable (le problème réel peut s'avérer inconsistant) alors que l'existence de mondes impossibles est plutôt

2. Il ne faut pas confondre $Bad(\mathcal{P})$ avec l'ensemble des mondes impossibles : ceux ci peuvent donner naissance à des CSP consistants comme inconsistants, qui ne nous intéressent *a priori* pas puisque, tant qu'il existe au moins un monde possible : le problème réel est forcément l'un des problèmes induit par un monde possible.

confortable, puisque limitant le nombre de CSP candidats.

Enfin, si l'on a affaire à un CSP probabiliste, nous pouvons calculer pour chaque décision potentielle la probabilité qu'elle soit satisfaisante, c'est-à-dire la probabilité qu'elle couvre le monde réel :

Définition 8 (Probabilité qu'une décision soit solution) *La probabilité qu'une décision d soit solution est la probabilité de l'ensemble des mondes qu'elle couvre i.e.,* $PS(d) = Pr(Sol(\langle \Lambda, W, \mathcal{C}[d] \rangle))$.

Il faut remarquer que PS *n'est pas* une distribution de probabilité [3].

Nous pouvons donc calculer la probabilité que le monde réel soit couvert : c'est la probabilité de consistance du problème réel.

Définition 9 (probabilité de consistance) *On associe à un CSP probabiliste p la probabilité que le problème réel soit consistant :*

$$P_{Cons}(\mathcal{P}) = Pr(Good(\mathcal{P}))$$

Si $P_{Cons}(\mathcal{P}) = 1$ *alors* $\mathcal{P}$ *sera dit* consistant.

Exemple (inspiré de [**?**] avec quelques modifications) : Considérons qu'il faut organiser un dîner, où aucun des trois invités (Grandgousier, Gargantua et Pantagruel) n'est sûr de venir. Il faut néanmoins choisir un vin et un plat principal, en fonction de quatre contraintes — trois d'entre elles dépendant de la présence des invités. Le CSP probabiliste $\mathcal{P}$ correspondant à ce problème est défini par :

- Variables de décision : $X = \{x_1, x_2\}$; $D_1 = \{\textsf{Blanc}, \textsf{Rouge}\}$; $D_2 = \{\textsf{Volaille}, \textsf{Bœuf}, \textsf{Poisson}\}$
- Variables contingentes : $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$; $W_1 = W_2 = W_3 = \{\textsf{présent}, \textsf{absent}\}$
- Contraintes : $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$;

De plus, les événements correspondant à la participation au repas des différents invités sont mutuellement indépendants. La probabilité que Grandgousier vienne est 0.6, et respectivement 0.9 et 0.5 pour Gargantua et Pantagruel.

3. C'est en fait une « *contour function* » au sens de la théorie de Dempster-Shafer.

Règles de bon goût

C_1 :

x_1	x_2
Blanc	*Poisson*
Rouge	*Bœuf*
Blanc	*Volaille*
Rouge	*Volaille*

Grandgousier n'aime pas le bœuf

C_2 :

λ_1	x_2
présent	*Volaille*
présent	*Poisson*
absent	*Volaille*
absent	*Poisson*
absent	*Bœuf*

Gargantua n'aime que le Rouge

C_3 :

λ_2	x_1
présent	*Rouge*
absent	*Rouge*
absent	*Blanc*

Pantagruel (végétarien !)

C_4 :

λ_3	x_2
présent	*Poisson*
absent	*Volaille*
absent	*Poisson*
absent	*Bœuf*

À partir de maintenant on abrégera *présent* et *absent* respectivement par p et a. Dans cet exemple, $C^* = \{C_1\}$. Tous les mondes sont possibles ($Poss(\mathcal{P}) = W$). Si l'on considère le monde $\omega_1 = (p, a, p)$, dont la probabilité est 0.03, la réduction des contraintes est : $Reduce(C_1, \omega_1) = C_1$, $Reduce(C_2, \omega_1) = \{\textit{Volaille}, \textit{Poisson}\}$, $Reduce(C_3, \omega_1) = \{\textit{Rouge}, \textit{Blanc}\}$ et $Reduce(C_3, \omega_1) = \{\textit{Poisson}\}$. Le problème $\langle X, D, \mathcal{C}[\omega_1]\rangle$ correspondant à ω_1 a une solution unique ($\textit{Blanc}, \textit{Poisson}$) et ω_1 est donc un monde sympathique. La décision ($\textit{Blanc}, \textit{Poisson}$) couvre trois autres mondes $\{(a, a, p), (p, a, a), (a, a, a)\}$, ce qui donne $PS((\textit{Blanc}, \textit{Poisson})) = 0.1$ (puisque les mondes couverts par ($\textit{Blanc}, \textit{Poisson}$) sont exactement ceux où Gargantua ne vient pas).

Dans le monde $\omega_2 = (p, p, p)$, dont la probabilité est 0.27, la réduction des contraintes est : $Reduce(C_1, \omega_2) = C_1$, $Reduce(C_2, \omega_2) = \{\textit{Volaille}, \textit{Poisson}\}$, $Reduce(C_3, \omega_2) = \{\textit{Rouge}\}$ et $Reduce(C_3, \omega_2) = \{\textit{Poisson}\}$, ce qui définit un CSP inconsistant. Ainsi, ω_2 est un monde antipathique. Le seul autre monde antipathique est $\omega_3 = (a, p, p)$, avec une probabilité de 0.18. Par conséquent, $Bad(\mathcal{P}) = \{(p, p, p), (a, p, p)\}$, $Pr(Bad(\mathcal{P}) = 0.27 + 0.18 = 0.45$ et $P_{Cons}(\mathcal{P}) = 0.55$.

2.4 Cas particuliers intéressants

Plusieurs cas particuliers intéressants de CSP mixtes et probabilistes peuvent être obtenus en ajoutant certaines hypothèses simplificatrices.

2.4.1 Indépendance mutuelle des variables contingentes

La première hypothèse simplificatrice correspond à l'indépendance mutuelle des variables contingentes. Elle s'exprime différemment selon qu'il s'agit de CSP mixtes (indépendance logique, ou structurelle, notée IS) ou de CSP probabilistes (indépendance probabiliste, IP) :

– le CSP mixte $\mathcal{P}$ vérifie l'hypothèse IS si et seulement si ses variables contingentes ne sont liées par aucune contrainte, c'est-à-dire que K ne contient que des contraintes unaires. Sans perte de généralité, on peut supposer que les domaines des variables contingentes tiennent déjà compte de ces valeurs impossibles, qui ont été au préalable retirées des domaines. On a alors $K = \varnothing$.

– le CSP probabiliste $\mathcal{P}$ vérifie l'hypothèse IP si ses variables contingentes sont mutuellement indépendantes au sens probabiliste du terme, c'est-à-dire $\forall \omega = (l_1, \ldots, l_p) \in W, pr(\omega) = \Pi_{i=1}^{p} pr_{\lambda_i}(l_i)$. Dans ce cas pr peut donc être spécifiée par une distribution de probabilité individuelle pour chaque variable contingente.

Notons que les hypothèses d'indépendance entraînent que le problème peut être écrit de façon à ce que $Poss(\mathcal{P}) = W$ (les valeurs impossibles ayant été retirées au préalable des domaines des variables contingentes).

2.4.2 *Variables contingentes booléennes et contraintes potentielles*

Cette seconde hypothèse simplificatrice, notée (B), est illustrée par l'exemple précédent : tandis que la contrainte de décision classique C_1 (*règles de bon goût*) fait indiscutablement partie du problème réel, chacune des autres contraintes est induite par une *contrainte potentielle* : par exemple, C_2 exprime que « si Grandgousier vient, alors la contrainte de décision classique c_2 : $Plat \in \{\textsf{Poisson}, \textsf{Volaille}\}$ doit être satisfaite ». Nous avons utilisé un paramètre booléen (λ_1) pour « contrôler » la contrainte potentielle c_2 correspondante. Plus généralement, un CSP mixte ou probabiliste $\mathcal{P} = \langle \Lambda, W, X, D, \mathcal{C}, K/pr \rangle$ est un *problème à contraintes potentielles* si $\mathcal{C} = \mathcal{C}^* \cup \{C_1, \ldots, C_q\}$, où :

– $\mathcal{C}^*$ est un ensemble de contraintes de décision classiques ne faisant intervenir que des variables de décision (contraintes *certaines*) ;

– $\{C_1, \ldots, C_q\}$ est induit par un ensemble de contraintes de décision classiques $\{c_1, \ldots, c_q\}$ (ne faisant intervenir que des variables de décision) ; les c_i sont des contraintes de décision *potentielles*, et chacune est codée dans $\mathcal{P}$ par une contrainte mixte C_i liant une ou plusieurs variables de décision (celles de c_i) à une variable contingente booléenne λ_i (dans notre exemple les deux valeurs possibles de ces variables contingentes sont *présent* et *absent*). La contrainte mixte C_i exprime exactement « si λ_i =*présent* alors c_i ».

– Si on ajoute à (B) l'hypothèse d'indépendance (IS/IP), alors on peut montrer que $Candidats(\mathcal{P})$ est un treillis de relaxation au sens de Freuder [FRE 92], éventuellement équipé de la distribution de probabilité induite de façon triviale par la distribution de probabilité sur les mondes (voir [FAR 93]). Plus précisément, un problème candidat est obtenu en ajoutant à $\mathcal{C}^*$ un sous-ensemble quelconque de $\{c_i, i = 1 \ldots q\}$, c'est-à-dire $Candidats(\mathcal{P}) = \{\langle X, D, \{c_i, i \in J\} \cup \mathcal{C}^* \mid J \subseteq \{1 \ldots q\}\rangle\}$.

2.4.3 Familles disjonctives de contraintes

Cette dernière classe de CSP mixtes / probabilistes, plus générale que la précédente, suppose que chaque contrainte mixte implique exactement une variable contingente, dont le domaine peut être de taille quelconque (la variable contingente n'est pas forcément booléenne). Chaque contrainte mixte correspond alors à une famille disjonctive de contraintes (une pour chaque valeur possible de la variable contingente concernée) [4]. Cette hypothèse, notée (F), généralise (B). Voir [**?**] pour plus de détails.

2.5 Décisions et décisions conditionnelles

Revenons aux situations définies en introduction concernant les connaissances que l'agent peut acquérir avant d'avoir à agir. Nous allons voir que selon qu'on se situe dans le cas (ACI) ou (API), les réponses souhaitables que le CSP mixte ou probabiliste doit donner à l'agent (en d'autres termes, les *solutions*) vont correspondre respectivement aux *décisions conditionnelles* et aux *décisions*.

2.5.1 Où les solutions sont des décisions conditionnelles

Dans cette sous-section nous faisons l'hypothèse que l'agent sera complètement informé de la valeur des variables contingentes (il connaîtra le monde réel) avant de prendre sa décision (**ACI**). Il suffira alors de donner à l'agent un comportement à suivre pour chaque monde possible : on peut alors construire une *décision conditionnelle* de $\mathcal{P}$ comme une fonction associant une décision à chaque monde possible.

Idéalement, une décision conditionnelle s associera à chaque monde possible une décision satisfaisant le problème de décision correspondant. Toutefois, comme certains mondes (les antipathiques) peuvent induire un problème de décision inconsistant, une requête plus raisonnable consistera à demander une *décision conditionnelle partielle*, c'est-à-dire définie sur un sous-ensemble $W(s)$ des mondes possibles (le mieux que l'on puisse faire est $W(s) = Good(\mathcal{P})$).

Définition 10 (décisions conditionnelles) *Une décision conditionnelle complète s est une fonction de $Poss(\mathcal{P})$ dans D. Une décision conditionnelle partielle s est une fonction d'un sous-ensemble $W(s)$ de $Poss(\mathcal{P})$ dans D. Une décision conditionnelle (partielle ou complète) est correcte si et seulement si $\forall \omega \in W(s), s(\omega)$ couvre ω.*

Dans le reste de l'article, nous utiliserons la terminologie « décisions conditionnelles » qu'il s'agisse de décisions complètes ou partielles. De façon claire, les décisions conditionnelles complètes généralisent les décisions comme on les a définies en Section 2. En fait, une décision conditionnelle constante sur $Poss(\mathcal{P})$ correspond à une décision. Les décisions conditionnelles correctes sont celles qui nous intéressent puisque toutes les décisions qu'elles donnent sont adéquates. Ce sont elles qui vont jouer le rôle des solutions pour les CSP mixtes et probabilistes.

4. Il est possible que certaines variables contingentes ne soient impliquées dans aucune contrainte mixte (ce sont alors des variables intermédiaires), ou dans plusieurs (la même variable contingente contrôle alors plusieurs familles).

Comme pour les décisions, on peut définir l'ensemble des mondes couverts par une décision conditionnelle, ainsi que (dans le cas probabiliste) la probabilité qu'une décision conditionnelle couvre le monde réel.

Définition 11 *L'ensemble des mondes couverts par la décision conditionnelle s est défini par :*

$$Cov(s) = \{\omega \in W(s) \mid s(\omega) \text{ couvre } \omega\}$$

La probabilité que s couvre le monde réel (quand c'est applicable) est définie par :

$$PS(s) = \sum_{\omega \in Cov(s)} pr(\omega)$$

Le cas idéal se présente lorsqu'il existe une décision conditionnelle couvrant tous les mondes possibles (une telle décision conditionnelle étant nécessairement correcte). Il est alors garanti qu'il existe une décision adéquate quel que soit le monde possible. On dira alors que $\mathcal{P}$ est conditionnellement consistant :

Définition 12 *$\mathcal{P}$ est conditionnellement consistant si et seulement si il existe une décision conditionnelle s telle que $Cov(s) = Poss(\mathcal{P})$.*

Proposition 1 *Les résultats suivants sont faciles à prouver :*

1. *pour toute décision conditionnelle s, $Cov(s) \subseteq Good(\mathcal{P}) \subseteq Poss(\mathcal{P})$, et $PS(s) \leq P_{Cons}(\mathcal{P})$;*
2. *il existe une décision conditionnelle s telle que $Cov(s) = Good(\mathcal{P})$ (et donc telle que $PS(s) = P_{Cons}(\mathcal{P})$) ;*
3. *la consistance conditionnelle de $\mathcal{P}$ est équivalente à $Good(\mathcal{P}) = Poss(\mathcal{P})$ ou encore à $Bad(\mathcal{P}) = \varnothing$.*

La définition suivante ne s'applique qu'aux CSP probabilistes.

Définition 13 (décisions conditionnelles optimales)) *Soit $\mathcal{P}$ un CSP probabiliste. Une décision conditionnelle s (partielle ou complète) est optimale pour $\mathcal{P}$ si et seulement si elle est correcte et elle maximise $PS(s)$ (ou de manière équivalente si et seulement si $PS(s) = P_{Cons}(\mathcal{P})$).*

Ainsi, on peut calculer une décision conditionnelle optimale « off line » et l'utiliser plus tard, en temps réel, une fois que le monde réel sera connu.
Exemple : reprenons le CSP probabiliste de la section précédente. On peut considérer la décision conditionnelle s suivante, qui est correcte, et optimale puisque $PS(s) =$

$P_{Cons}(\mathcal{P}) = 0.55$[5].

λ_1	λ_2	λ_3	Décision
p	p	a	(*Rouge*, *Volaille*)
p	a	a	(*Rouge*, *Volaille*
a	p	a	(*Rouge*, *Volaille*)
a	a	a	(*Rouge*, *Volaille*)
p	a	p	(*Blanc*, *Poisson*)
a	a	p	(*Blanc*, *Poisson*)

2.5.2 Où les solutions ne sont pas conditionnées aux mondes

Dans la situations contraire (**API**) — l'agent n'apprendra jamais rien d'autre sur le monde réel —, chercher une décision conditionnelle ne sert à rien puisque l'agent sera incapable de l'instancier correctement. Dans ce cas, il faut donc chercher une simple *décision* plutôt qu'une décision conditionnelle. Le cas idéal se présente quand il existe une décision couvrant tous les mondes possibles.

Définition 14 (décisions universelles, consistance universelle) *Une décision d est une décision universelle pour $\mathcal{P}$ si et seulement si $Cov(d) \supseteq Good(\mathcal{P})$. Elle est dite universelle forte si $Cov(d) = Poss(\mathcal{P})$. $\mathcal{P}$ est dit fortement consistant s'il possède une décision universelle forte.*

Une décision universelle forte est une décision couvrant *tous* les mondes possibles. Dans le cas probabiliste, une telle décision vérifie $PS(d) = 1$. La consistance forte constitue un cas idéal. Il est clair que la consistance forte implique la consistance conditionnelle.

Notons que le cas de la décision « simplement » universelle semble le plus pertinent : une telle décision conviendra à chaque fois que c'est possible, c'est-à-dire pour chaque monde possible dont le problème de décision associé est consistant — et il est dans ce cas inutile de chercher une décision conditionnelle (qui ne ferait pas mieux qu'une décision universelle).

Dans le cas d'un CSP probabiliste, quand $\mathcal{P}$ n'est pas fortement consistant, une stratégie raisonnable consiste à chercher une décision maximisant la probabilité d'être solution[6] (cf. Définition 8) :

Définition 15 (décisions optimales) *Une décision d est optimale si et seulement si elle maximise $PS(d)$. On pose $P^*(\mathcal{P}) = PS(d)$ où d est une décision optimale.*

$P^*(\mathcal{P})$ est la probabilité maximale de succès d'une décision. Il est évident que $P^*(\mathcal{P}) \leq P_{Cons}(\mathcal{P})$. L'égalité $PS(d) = P_{Cons}(\mathcal{P})$ est atteinte lorsque $\mathcal{P}$ possède une décision universelle. Notons qu'une décision universelle est nécessairement optimale.
Exemple (suite) : La décision (*Rouge*, *Volaille*) est la seule décision optimale (on a $PS((Rouge, Volaille)) = 0.5 = P^*(\mathcal{P})$) mais elle n'est pas universelle forte et le problème n'est donc pas fortement consistant. Toutefois, si la probabilité que Pantagruel vienne ($\lambda_3 = p$) était nulle, (*Rouge*, *Volaille*) aurait été une décision universelle forte.

---

5. La valeur de s est indéfinie sur les mondes antipathiques de $Bad(\mathcal{P})$.
6. Dans le cas des CSP mixtes, une définition similaire consisterait à considérer des décisions s telles que $Cov(s)$ soit maximal au sens de l'inclusion.

3 Recherche d'une décision optimale pour un CSP probabiliste

Dans les sections 3 et 4, nous considérons des CSP probabilistes — toutefois des algorithmes similaires à ceux que nous allons donner peuvent être définis dans le cadre des CSP mixtes (voir [**?**] pour le calcul d'une décision conditionnelle). Ensuite, dans ces deux sections, nous allons supposer (pour des raisons de simplicité) que **les variables contingentes sont mutuellement indépendantes (IP)**. Cependant, cette hypothèse pourrait être relâchée sans encombre pourvu que l'on sache calculer la probabilité d'un ensemble de mondes.

Dans cette section (3), on fait de plus l'hypothèse qu'on est dans la situation **(API)** *i.e.,* qu'avant d'agir l'agent ne connaîtra rien de plus sur le monde réel (tout ce qu'il sait est contenu dans pr) et, par conséquent, tout ce qu'il peut faire est d'exécuter une décision. Idéalement, l'agent aimerait que le résolveur lui fournisse une décision *optimale* mais le temps de calcul pour l'obtenir peut être prohibitif en présence d'une échéance impérative. On présente pour ce cas un algorithme *anytime* de calcul d'une décision qui est d'autant meilleure (dans le sens du critère d'optimalité) qu'on lui permet de tourner longtemps. La décision obtenue a la propriété d'être optimale si on laisse l'algorithme se dérouler jusqu'à son arrêt naturel.

Notre approche de ce problème de maximisation de $PS(d)$ repose sur un algorithme de recherche de type *Branch and Bound* en profondeur d'abord. Dans un but de clarté, supposons que les variables sont instanciées dans l'ordre $(x_1, ..., x_n)$ et que toutes les contraintes sont binaires[7]. La racine de l'arbre de recherche est l'instanciation vide. Les nœuds intermédiaires représentent des décisions partielles $(d_1, ..., d_i)$. Les feuilles représentent des instanciations de X *i.e.*, des décisions $d = (d_1, ..., d_n)$. Dans l'exploration en profondeur d'abord de l'arbre de recherche, on garde une trace de la dernière feuille trouvée maximisant $PS(d)$.

Chaque fois qu'une nouvelle valeur d_i est affectée à la variable x_i, toutes les contraintes $C \in \mathcal{C}$ liant cette variable x_i à une variable de décision x_j ou à une variable contingente λ_k sont exploitées par un algorithme classique de type *Forward-checking* [HAR 80] pour éliminer toutes les valeurs incompatibles avec l'affectation $x_i = d_i$ du domaine de x_i (resp. λ_k) : chaque domaine D_j (resp. W_k) est simplement remplacé par $D_j \cap Reduce(C, (d_i))$ (resp.$W_k \cap Reduce(C, (d_i))$ si C implique une variable contingente). Cette mise à jour des domaines suite à l'affectation de la valeur d_i à x_i est réalisée par la procédure ForwardCheck dans l'algorithme 1 (ligne 2).

Après la propagation via la procédure ForwardCheck, le produit cartésien des valeurs restantes dans les domaines des variables contingentes définit les mondes qui sont compatibles avec la décision partielle courante : tous les mondes qui sont manifestement incompatibles ont été écartés. Le cumul des probabilités des mondes restants donne un majorant $\widehat{PS}(d)$ de la probabilité de la meilleure décision complète parmi les descendants du nœud courant puisque seuls des mondes incompatibles ont été éliminés. Ce majorant est facilement calculé si l'indépendance est supposée : c'est le produit sur tous les domaines de la somme des probabilités des valeurs restantes de

7. Ceci implique que nous soyons dans le cas (F) parce qu'une contrainte doit contenir au moins une variable de décision. L'algorithme est facilement généralisable à des contraintes n-aires tant que l'hypothèse (F) est vérifiée.

chaque domaine. Lorsqu'une décision complète d est atteinte, toutes les contraintes ayant été propagées, les mondes restants sont les mondes couverts par la décision d et $\widehat{PS}(d) = PS(d)$.

Quand l'algorithme doit-il effectuer un retour-arrière ? Si après réalisation d'un *forward-checking*, le domaine d'une variable de décision devient vide ou si le domaine d'une variable contingente devient vide (ce qui se traduit par $p_{i+1} = 0$), il est clair qu'aucun monde ne peut être couvert par la décision courante. Le retour-arrière qui s'impose alors consiste à remettre les domaines dans l'état où ils étaient avant la dernière affectation et à poursuivre la recherche en essayant une nouvelle valeur pour la variable de décision infructueusement affectée. Une autre situation de retour-arrière se produit lorsque le majorant atteint ou passe au dessous du seuil α qui joue le rôle de minorant dans le *Branch and Bound*. α est simplement la probabilité $PS(d)$ associée à la meilleure décision trouvée jusque là. Ce seuil est initialisé à 0 et mis à jour chaque fois qu'une décision d telle que $PS(d) > \alpha$ est atteinte. Lorsque minorant et majorant se rencontrent, on sait que la probabilité de toute décision complète descendant de ce nœud ne sera pas supérieure à celle de la meilleure décision déjà disponible. Il est donc futile de chercher à étendre cette décision partielle, d'où un retour-arrière.

L'algorithme s'arrête dès qu'aucun nouveau nœud ne peut être généré. Il réussit lorsqu'au moins une feuille est produite. S'il n'est pas interrompu, la meilleure décision trouvée est nécessairement optimale.

Function Cherche$(d_1, \ldots, d_i, p_i)$
si $i = n$ **alors** $\alpha := p_i$ $\{$Mise à jour de l'évaluation$\}$
sinon
  Choisir une variable x_{i+1}
  pour $d_{i+1} \in D_{i+1}$ **faire**
    SauveDomaines$(\Lambda \cup X)$
    pour $C \in \mathcal{C}$ t.q. $x_{i+1} \in V(C)$*et* $V(C) \not\subseteq \{x_1, \ldots, x_{i+1}\}$ **faire**
1   Let $\{v_j\} = V(C) - \{x_{i+1}\}$
2   ForwardCheck(v_j, C, d_{i+1})
3    $p_{i+1} := \prod_{W_k} \sum_{v \in W_k} [pr_{\lambda_k}(v)]$
    si $(p_{i+1} > \alpha)$ *et* $(\forall i, D_i \neq \varnothing)$ **alors**
      Cherche$(d_1, \ldots, d_{i+1}, p_{i+1})$
    RestaureDomaines$(\Lambda \cup X)$

Algorithme 1: *Recherche d'une décision optimale*

L'algorithme de recherche repose sur la fonction Cherche dont l'appel initial doit se faire avec une décision partielle vide et une probabilité p_0 égale à 1,q majorant initial de la probabilité d'une décision optimale. La fonction SauveDomaines sauvegarde les domaines des variables (en fait, seuls les domaines modifiés doivent être sauvegardés). La fonction RestaureDomaines restitue les domaines sauvegardés.

Il est intéressant de remarquer qu'à la ligne marquée 1 de l'algorithme, v_j peut être aussi bien une variable de décision qu'une variable contingente. À la ligne marquée 3, la valeur courante de la borne supérieure p_i pourrait facilement être mise à jour incrémentalement en prenant en compte seulement les variables contingentes dont le domaine a changé : il suffirait de multiplier la probabilité précédente p_i par la diminution relative de la probabilité associée à chaque domaine modifié. Cet algorithme s'apparente à des extensions du *Forward checking* utilisées dans le cadre des CSP partiels [FRE 92] et des CSP valués [SCH 95, SCH re].

4 Recherche d'une décision conditionnelle pour un CSP probabiliste

Dans cette section on se place dans le cas où l'agent sera complètement informé du monde réel avant d'agir (**ACI**).

Il est clair que la recherche d'une décision conditionnelle optimale a un coût calculatoire élevé [8] ; cela nous mène à chercher un algorithme qui maintient à tout instant une décision conditionnelle qui tend à être optimale si on laisse l'algorithme se dérouler jusqu'à ce qu'il s'arrête naturellement (plus on le laisse se dérouler, plus la probabilité associée à la décision conditionnelle courante est élevée). Intuitivement, l'algorithme construit incrémentalement une décision conditionnelle qui, à la fin, couvrira un sur-ensemble [9] de tous les mondes sympathiques. Il consiste à répéter la suite d'opérations suivantes :

1. choisir une nouvelle décision d (à ajouter à la décision conditionnelle partielle courante) qui couvre au moins un monde possible parmi ceux qui ne sont pas encore couverts ;
2. calculer l'ensemble R des mondes que cette décision couvre, et
3. soustraire cet ensemble de l'ensemble des mondes qui n'ont pas encore été couverts (cet ensemble étant initialisé à W, c'est-à-dire tous les mondes — possibles ou non).

Pour calculer facilement les mondes couverts par d et leurs probabilités, on fait les deux hypothèses suivantes : d'abord, les variables contingentes sont mutuellement indépendantes (IP), et ensuite, toute contrainte de C implique *au plus une variable contingente* (F). L'hypothèse (F) implique que les mondes couverts par une décision d, c'est-à-dire $C[d]$, forment un produit cartésien de sous-ensembles des domaines des variables contingentes. Les soustractions successives de ces produits cartésiens sont effectuées en utilisant une technique récemment proposée par Freuder et Hubbe [FRE 95], appelée *extraction de sous-domaines*. Avant de décrire notre algorithme, nous rappelons d'abord cette technique.

4.1 Extraction de sous-domaines

On définit un *environnement* E comme une ensemble de mondes de la forme $\ell_1 \times \cdots \times \ell_p$, avec $\forall k, \ell_k \subseteq W_k$. Un exemple d'environnement est $(\lambda_1 \in \{p\}, \lambda_2 \in$

8. Puisque sa taille atteint, dans le pire des cas, le nombre de mondes possibles. On notera aussi que la complexité du problème de l'existence d'une décision conditionnelle complète est Π_2^p-complet [**?**].

9. Des mondes impossibles peuvent éventuellement se trouver couverts.

$p, a\}, \lambda_3 \in \{a\})$, qu'on peut aussi écrire $\{p\} \times \{p, a\} \times \{a\}$ ou encore $\left[\begin{smallmatrix} p \\ p,a \\ a \end{smallmatrix}\right]$ quand il n'y a aucune ambiguïté sur l'ordre des variables contingentes. L'ensemble de tous les environnements possibles est un treillis (équipé de l'ordre d'inclusion), dont le sommet est l'ensemble de toutes les affectations de variables contingentes (sur l'exemple $\left[\begin{smallmatrix} p,a \\ p,a \\ p,a \end{smallmatrix}\right]$) et dont la base est l'ensemble vide. Sous l'hypothèse d'indépendance mutuelle des variables contingentes, la probabilité d'un environnement $E = \ell_1 \times \cdots \times \ell_p$, est $Pr(E) = \prod_{k=1}^{p} \sum_{v \in \ell_k} pr_{\lambda_k}(v)$. Dans la suite, nous confondrons parfois, pour des raisons de facilité, un environnement E et l'ensemble des contraintes unaires sur les variables contingentes qui définissent E.

Étant donnés deux environnements E et F, la technique dite d'*extraction de sous-domaines* [FRE 95] décompose E en un ensemble de sous-environnements disjoints $\mathsf{Dec}(E, F)$ tels que tous les mondes de E appartiennent soit à F soit à l'un des sous-environnements de la décomposition. La décomposition est unique si l'ordre des variables est fixé. Nous donnons ici une version probabiliste de la décomposition de Freuder et Hubbe, comprenant, en plus de la technique originale, un calcul incrémental de la probabilité de chaque environnement généré par la décomposition. Cette fonction retourne la décomposition de E par F et la probabilité de l'ensemble de mondes de E qui sont déjà dans F.

En pratique, la probabilité p_E de l'environnement E dont on a besoin en entrée a déjà été calculée auparavant, puisqu'à chaque appel de la procédure Dec, soit $E = W$ et $p_E = 1$, soit E provient d'une décomposition précédente.

Notons que quand $E \cap F = \varnothing$ on a $\mathsf{Dec}(E, F) = \langle \{E\}, 0 \rangle$, et quand $E \subseteq F$ on a $\mathsf{Dec}(E, F) = \langle \varnothing, p_E \rangle$.

Exemple d'exécution de l'algorithme 2 (toujours sur le même exemple) :

$E = W = \left[\begin{smallmatrix} p,a \\ p,a \\ p,a \end{smallmatrix}\right], F = \left[\begin{smallmatrix} a \\ p,a \\ a \end{smallmatrix}\right]$.

1. $i = 1$; $Rest = \left[\begin{smallmatrix} p \\ p,a \\ p,a \end{smallmatrix}\right]$;$p_{Rest} = 0.6$;$E' = \left[\begin{smallmatrix} a \\ p,a \\ p,a \end{smallmatrix}\right]$; $p_{E'} = 0.4$; $\mathtt{List} = \{\langle \left[\begin{smallmatrix} p \\ p,a \\ p,a \end{smallmatrix}\right], 0.6 \rangle\}$;

2. $i = 2$; $Rest = \left[\begin{smallmatrix} a \\ \\ p,a \end{smallmatrix}\right]$; $p_{Rest} = 0$;

3. $i = 3$; $Rest = \left[\begin{smallmatrix} a \\ p,a \\ p \end{smallmatrix}\right]$; $p_{Rest} = 0.2$; $E' = \left[\begin{smallmatrix} a \\ p,a \\ a \end{smallmatrix}\right]$; $p(E') = 0.2$; $\mathtt{List} = \{\langle \left[\begin{smallmatrix} p \\ p,a \\ p,a \end{smallmatrix}\right], 0.6 \rangle, \langle \left[\begin{smallmatrix} a \\ p,a \\ p \end{smallmatrix}\right], 0.2 \rangle\}$. FIN

4.2 Un algorithme de recherche d'une décision conditionnelle

Deux étapes de l'algorithme 3 méritent des commentaires : d'abord (1), les séquences de CSP $\langle \Lambda \cup X, W \times D, C \cup E \rangle$, résolus successivement, définissent des problèmes de satisfaction de contraintes *dynamiques* [DEC 88, VER 95] successifs, dont la résolution peut largement bénéficier des techniques développées pour résoudre de tels problèmes. Ensuite (2), une fois qu'un nouveau WC est construit, il est soustrait de tous les environnements en attente pour éviter des calculs redondants. Cela

Fonction $\mathsf{Dec}(\langle E, p_E \rangle, F)$

$\mathtt{List} := \varnothing$; $E' := E$; $p_{E'} := p_E$; $i := 1$
répéter
  $Rest := E'$
  $Rest_{\downarrow\{\lambda_i\}} := E'_{\downarrow\{\lambda_i\}} - F_{\downarrow\{\lambda_i\}}$
  $p_{Rest} := p_{E'} \cdot \frac{p(Rest_{\downarrow\{\lambda_i\}})}{p(E'_{\downarrow\{\lambda_i\}})}$ {indépendance}
  si $(Rest_{\downarrow\{\lambda_i\}} \neq \varnothing)$ **alors**
    Add $\langle Rest, p_{Rest} \rangle$ to $\mathtt{List}$
    $E'_{\downarrow\{\lambda_i\}} := E'_{\downarrow\{\lambda_i\}} \cap F_{\downarrow\{\lambda_i\}}$
    $p_{E'} := p_{E'} - p_{Rest}$
  $i := i + 1$
jusqu'à *($(i > p)$* **or** *$(E'_{\downarrow\{\lambda_{i-1}\}} = \varnothing)$)*
retourner $\langle \mathtt{List}, p_{E'} \rangle$

Algorithme 2: *L'algorithme de décomposition*

$\mathtt{Decisions} := \varnothing$ {décisions & environnements couverts}
$\mathtt{Env} := \{\langle W, 1 \rangle\}$ {environnements non couverts}
$\mathtt{Bad} := \varnothing$ {environnements non couvrables}
$p_{good} := 0$ {minorant de $P_{Cons}(\mathcal{P})$}
$p_{bad} := 0$ {minorant de $1 - P_{Cons}(\mathcal{P})$}
répéter
  Prendre une paire $\langle E, p_E \rangle$ dans $\mathtt{Env}$ {heuristique possible}
  si $\langle \Lambda \cup X, W \times D, \mathcal{C} \cup E \rangle$ *est inconsistant* **alors**
    %E ne contient que des mondes non couvrables
    $\mathtt{Bad} := \mathtt{Bad} \cup E$; $p_{bad} := p_{bad} + p_E$
  sinon
    $(d, \omega) :=$ une solution de $\langle \Lambda \cup X, W \times D, \mathcal{C} \cup E \rangle$
    %d couvre au moins un monde possible de E
    $WC := \mathcal{C}[d]$ {mondes couverts par d }
    Ajouter $\langle WC, d \rangle$ à $\mathtt{Decisions}$;
    $\mathtt{Env}' = \varnothing$
    pour $G \in \mathtt{Env}$ **faire**
      $\langle H, p \rangle := \mathsf{Dec}(G, WC)$
      $\mathtt{Env}' := \mathtt{Env}' \cup H$
      $p_{good} := p_{good} + p$
    $\mathtt{Env} := \mathtt{Env}'$
jusqu'à *($\mathtt{Env} = \varnothing$) (ou interruption par l'utilisateur)*

Algorithme 3: *Calcul d'une décision conditionnelle optimale* garantit en outre que l'algorithme s'arrêtera quand la liste courante `Decisions` définit la solution.

Exemple d'exécution de l'algorithme 3 (toujours sur le même exemple) :

1. $\mathtt{Env} = \{\langle W, 1\rangle\}$; $p_{good} = 0$; $p_{bad} = 0$; $\mathtt{Decisions} = \varnothing$

2. $\langle E, p_E\rangle = \langle W, 1\rangle$; $d_1 = (R, B)$; $WC = \left[\begin{smallmatrix} a \\ p,a \\ a \end{smallmatrix}\right]$;
   $\mathtt{Env} = \{\langle \left[\begin{smallmatrix} p \\ p,a \\ p,a \end{smallmatrix}\right], 0.6\rangle, \langle \left[\begin{smallmatrix} a \\ p,a \\ p \end{smallmatrix}\right], 0.2\rangle\}$;
   $\mathtt{Decisions} = (\langle \left[\begin{smallmatrix} a \\ p,a \\ a \end{smallmatrix}\right], (\mathit{Rouge}, \mathit{Bœuf})\rangle)$; $p_{good} = 0.2$.

3. $\langle E, p_E\rangle = \langle \left[\begin{smallmatrix} p \\ p,a \\ p,a \end{smallmatrix}\right], 0.6\rangle$; $d_2 = (\mathit{Rouge}, \mathit{Volaille})$; $WC = \left[\begin{smallmatrix} p,a \\ p,a \\ a \end{smallmatrix}\right]$;
   $\mathtt{Env} = \{\langle \left[\begin{smallmatrix} p \\ p,a \\ p \end{smallmatrix}\right], 0.3\rangle, \langle \left[\begin{smallmatrix} a \\ p,a \\ p \end{smallmatrix}\right], 0.2\rangle\}$;
   $\mathtt{Decisions} = (\langle \left[\begin{smallmatrix} p,a \\ p,a \\ a \end{smallmatrix}\right], (\mathit{Rouge}, \mathit{Volaille})\rangle, \langle \left[\begin{smallmatrix} a \\ p,a \\ a \end{smallmatrix}\right], (\mathit{Rouge}, \mathit{Bœuf})\rangle)$;
   $p_{good} = 0.5$.

4. $\langle E, p_E\rangle = \langle \left[\begin{smallmatrix} p \\ p,a \\ p \end{smallmatrix}\right], 0.3\rangle$; $d_3 = (\mathit{Blanc}, \mathit{Poisson})$; $WC = \left[\begin{smallmatrix} p,a \\ a \\ p,a \end{smallmatrix}\right]$;
   $\mathtt{Env} = \{\langle \left[\begin{smallmatrix} p \\ p \\ p \end{smallmatrix}\right], 0.27\rangle, \langle \left[\begin{smallmatrix} a \\ p \\ p \end{smallmatrix}\right], 0.18\rangle\}$;
   $(\langle \left[\begin{smallmatrix} p,a \\ a \\ p,a \end{smallmatrix}\right], (\mathit{Blanc}, \mathit{Poisson})\rangle$ est ajouté à `Decisions` ; $p_{good} = 0.55$;

5. $\langle E, p_E\rangle = \langle \left[\begin{smallmatrix} p \\ p \\ p \end{smallmatrix}\right], 0.27\rangle$: inconsistant ;
   $\mathtt{Bad} = \{\left[\begin{smallmatrix} p \\ p \\ p \end{smallmatrix}\right]\}$; $p_{bad} = 0.27$; $\mathtt{Env} = \{\langle \left[\begin{smallmatrix} a \\ p \\ p \end{smallmatrix}\right], 0.18\rangle\}$.

6. $\langle E, p_E\rangle = \langle \left[\begin{smallmatrix} a \\ p \\ p \end{smallmatrix}\right], 0.18\rangle$: inconsistant ;
   $\mathtt{Bad} = \{\left[\begin{smallmatrix} p \\ p \\ p \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ p \\ p \end{smallmatrix}\right]\}$; $p_{bad} = 0.45$; $\mathtt{Env} = \varnothing$. FIN.

$(\langle \left[\begin{smallmatrix} p,a \\ a \\ p,a \end{smallmatrix}\right], (\mathit{Blanc}, \mathit{Poisson})\rangle, \langle \left[\begin{smallmatrix} p,a \\ p,a \\ a \end{smallmatrix}\right], (\mathit{Rouge}, \mathit{Volaille})\rangle, \langle \left[\begin{smallmatrix} a \\ p,a \\ a \end{smallmatrix}\right], (\mathit{Rouge}, \mathit{Bœuf})\rangle)$ est la valeur finale de `Decisions` à partir de laquelle on peut maintenant construire une décision conditionnelle ; de plus $Bad(\mathcal{P})$ est contenu dans `Bad`. p_{good} tend vers $P_{Cons}(\mathcal{P})$, qu'il atteint à la fin de l'algorithme.

4.3 Correction de l'algorithme

Proposition 2 *En tout point de l'algorithme 3, les propriétés suivantes sont vérifiées :*

- $\forall \langle E, p_E\rangle$ *de* `Env` *on a* $p_E = Pr(E)$
- $\forall E \in$ `Bad` *on a* $E \subseteq Bad(\mathcal{P})$ *;*
- $\forall \langle E, d\rangle \in$ `Decisions`, $\forall \omega \in E, d$ *couvre* ω *;*
- $p_{good} \leq Pr(Good(P)) \leq 1 - p_{bad}$

Proposition 3 (Correction de l'algorithme) *Si on le laisse se dérouler sans l'interrompre, l'algorithme 3 s'arrête et :*

- *la valeur finale de* `Decisions` *définit une décision conditionnelle optimale ;*

– *les valeurs finales de p_{good} et p_{bad} vérifient :*

$$p_{good} + p_{bad} = 1, p_{good} = Pr(Good(\mathcal{P})),\ p_{bad} = Pr(Bad(\mathcal{P}))$$

Les preuves des quatre points de la Proposition 2 ne présentent pas de difficulté particulière; celle de la Proposition 3 repose essentiellement sur les deux invariants suivants, vérifiés en tout point de l'exécution de l'algorithme 2 :

– $\forall \omega \in W$, exactement l'une de ces trois assertions est vérifiée :
  (a) il existe un $\langle E, d\rangle$ dans `Decisions` tel que $\omega \in E$
  (b) il existe un unique E dans `Env` tel que $\omega \in E$
  (c) il existe un unique E dans `Bad` tel que $\omega \in E$

– $p_{good} + p_{bad} + \sum_{\langle E, p_E\rangle \in \texttt{Env}} p_E = 1$

Ainsi, l'ensemble de mondes couverts par `Decisions` croît de façon monotone (ainsi que `Bad`), et l'intervalle $[p_{good}, 1 - p_{bad}]$ se réduit de plus en plus de sa valeur initiale $[0, 1]$ jusqu'à sa valeur finale précise $[Pr(Good(\mathcal{P})), Pr(Good(\mathcal{P}))]$ (il est possible d'utiliser l'algorithme de façon « anytime »). `Decisions` tend vers une couverture de l'ensemble des mondes sympathiques $Good(\mathcal{P})$ et `Bad` tend vers $Bad(\mathcal{P})$.

Jusqu'à présent, nous avons supposé que l'applicabilité d'une décision conditionnelle ne posait pas de problème (puisque, selon l'hypothèse qu'on est dans la situation (ACI), le monde réel sera connu de l'agent avant qu'il n'ait à prendre sa décision). On pourrait considérer des situations plus complexes où l'agent serait en mesure d'acquérir certaines connaissances sur le monde par des « actions informatives » (en anglais « knowledge-gathering actions ») ; le problème consisterait alors à trouver un ensemble pertinent d'actions informatives, suffisant pour permettre à l'agent de prendre sa décision.

5 Travaux connexes

Dans les domaines des CSP et du raisonnement automatique : les CSP partiels [FRE 92] définissent un cadre général qui étend le modèle CSP traditionnel dans le même esprit que les CSP mixtes/probabilistes, mais sans toutefois faire une distinction entre variables de décision et variables contingentes. La résolution de CSP mixtes/probabilistes pourrait bénéficier de travaux faits pour les *CSP dynamiques* [DEC 88, VER 95], où plusieurs techniques sont proposées pour résoudre une séquence de CSP qui diffèrent l'un de l'autre seulement sur quelques contraintes. Ces techniques sont plus efficaces que l'approche naïve consistant à résoudre chaque CSP l'un après l'autre. Elles pourraient être incorporées dans l'algorithme de recherche d'une décision conditionnelle dans lequel se succède les résolutions des CSP$\langle \Lambda \cup X, W \times D, C \cup E\rangle$, E variant. D'autre part, une représentation compacte d'un ensemble de solutions (partielles) pourrait vraisemblablement être réalisée par un OBDD (Ordered Binary Decision Diagrams) [**?**] ou par un automate à états finis déterministe. Les OBDD sont utilisés pour résoudre des problèmes issus de la logique propositionnelle, proches des CSP. En fait, un OBDD pourrait — avec une complexité spatiale et temporelle exponentielle dans le pire des cas — représenter toutes les solutions d'un CSP mixte en se ramenant à une logique propositionnelle (voir [JÉG 94, JÉG 95] pour une utilisation des BDD pour le traitement de CSP dynamiques).

Dans les domaines touchant à la théorie de la décision : La prise de décision sous incertitude a été beaucoup étudiée dans le cadre de la théorie de la décision et commence à être appliquée en planification. Plusieurs approches de représentation des problèmes de décision font une distinction, d'une manière ou d'une autre, entre les variables contingentes (dont les valeurs sont en général soumises à des distributions de probabilité) et les variables de décision. C'est le cas, en particulier, des diagrammes d'influence [**?**], des processus de décision markoviens et des systèmes valués à la Shenoy [**?**]. D'un côté, ces cadres sont plus généraux que l'approche décisionnelle de cet article parce qu'ils peuvent s'appliquer à des problèmes de décision séquentiels et prennent en compte l'utilité des décisions. D'un autre côté, dans notre approche les liens entre les variables sont décrits à l'aide d'un langage de représentation équipé d'outils algorithmiques élaborés, alors que les diagrammes d'influence, par exemple, exigent un codage explicite assez lourd où l'interaction entre un jeu de variables de décisions et les variables contingentes doit être compilée dans une table de transition (une distribution de probabilité conditionnelle). Les deux types d'approches semblent assez complémentaires et devraient pouvoir se combiner de façon à tirer profit des avantages de chacune.

En représentation des connaissances : Récemment, Boutilier [**?**] a proposé une logique pour une version qualitative de la théorie de la décision, comprenant également une distinction entre variables contrôlables et incontrôlables ; on retrouve cette même distinction dans [**?**] ; toutefois, Boutilier ne s'intéresse qu'à la recherche d'une décision (celle satisfaisant au mieux le plus mauvais des cas possible) et ne considère donc pas la recherche de décisions conditionnelles. Enfin, dans [**?**] une distinction similaire entre les choix de la nature de ceux de l'agent (dans un cadre programmation logique) est faite dans l'optique de construire des stratégies qui sont assez similaires à nos décisions conditionnelles.

6 Conclusion

La contribution principale de cet article consiste en une extension du cadre CSP permettant de traiter des problèmes de décision sous incertitude, en distinguant les variables de décision contrôlables des variables contingentes qui ne le sont pas, et en représentant l'incertitude sur les valeurs des variables contingentes soit de manière qualitative par un ensemble de contraintes-connaissances (CSP mixtes), soit par une distribution de probabilité (CSP probabilistes). Pour les CSP mixtes probabilistes, qui présentent plus d'intérêt pratique, nous avons développé deux algorithmes de recherche de décisions, l'un applicable lorsque l'agent peut connaître le monde avant l'échéance à laquelle il est impératif d'agir, l'autre étant adapté aux situations où l'hypothèse précédente ne tient pas.

Cette extension constitue un premier pas vers l'intégration de la théorie de la décision dans l'approche CSP. Afin d'aller plus loin dans cette direction, deux étapes importantes seront nécessaires : il faudra d'une part, incorporer la notion d'utilité, et d'autre part, pouvoir traiter des problèmes de décision séquentiels. Les fonctions d'utilité sont un moyen de représenter la flexibilité et peuvent se coder dans notre modèle à l'intérieur des contraintes de $\mathcal{C}$, pour lesquelles on ajoute à chaque n-uplet un champ contenant l'utilité du n-uplet en question (voir [SCH 95, SCH re]). Dans une échelle donnée d'utilité, on attribuera l'utilité minimale aux n-uplets interdits. Une intégration de la flexibilité (modélisée par des contraintes floues) et de l'incertitude (probabiliste) dans les CSP a été fait dans [FAR 94] ; la notion d'utilité correspond dans ce cas à un degré de satisfaction et peut alors s'agréger avec l'incertitude probabiliste en considérant l'espérance de satisfaction. Une illustration d'application à un problème de planification agricole est donnée dans [**?**].

L'extension de notre approche aux problèmes de décisions séquentiels (ou multi-étapes) est plus ardue. La partition entre variables de décision et variables contingentes, telle que nous l'avons faite, n'est plus suffisante car non seulement les décisions doivent se prendre sous les restrictions imposées par les variables contingentes, mais les valeurs de certaines variables contingentes peuvent être influencées par des décisions antérieures. La représentation doit vraisemblablement être enrichie de manière à prendre en compte la structure orientée des dépendances entre les variables comme cela est fait dans un diagramme d'influence [**?**] ou un réseau causal [**?**].

Enfin, on remarque que de nombreux travaux récents en planification utilisent la théorie de la décision. Les développements futurs des CSP probabilistes pour prendre en compte les aspects utilité et séquences de décisions vont donc aussi dans cette direction ; notre objectif à long terme est de relier la satisfaction de contraintes et la planification basée sur la théorie de la décision.

Remerciements : Nous remercions Jean-Pierre RELLIER pour les discussions fructueuses que nous avons pu avoir et Christophe FIORIO pour son style `algorithm` (cf. `http://www.lirmm.fr/~fiorio/AlgorithmSty/presentation.html`).

Références

[BIS 95] BISTARELLI S., MONTANARI U. et ROSSI F., Constraint solving over semirings. *In : Proc. of the 14th IJCAI*, Montréal, Canada, août 1995.

[DEC 88] DECHTER R. et DECHTER A., Belief maintenance in dynamic constraint networks. *In : Proc. of AAAI-88*, p. 37–42, St. Paul, MN, 1988.

[FAR 93] FARGIER H. et LANG J., Uncertainty in constraint satisfaction problems: a probabilistic approach. *In : Proc. of ECSQARU '93, LNCS 747*, p. 97–104, Grenade, Spain, novembre 1993.

[FAR 94] FARGIER H., LANG J., MARTIN-CLOUAIRE R. et RELLIER J.-P., Uncertainty and flexibility in constraint satisfaction: A case study. *In : Proceedings ECAI'94 Workshop on Constraint Satisfaction Issues raised by Practical Applications*, éd. par SCHIEX T. et BESSIÈRE C., Amsterdam, août 1994.

[FRE 92] FREUDER E. et WALLACE R., Partial constraint satisfaction. *Artificial Intelligence*, vol. 58, décembre 1992, p. 21–70.

[FRE 95] FREUDER E., Using metalevel constraint knowledge to reduce constraint checking. *In : Constraint Processing: Selected papers from the ECAI'94 workshop*, éd. par MEYER M., Springer Verlag.

[HAR 80] HARALICK R. M. et ELLIOT G. L., Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, vol. 14, 1980, p. 263–313.

[JÉG 94] JÉGOU P., A logical approach to solve dynamic CSPs: Preliminary report. *In : Proceedings ECAI'94 Workshop on Constraint Satisfaction Issues raised by Practical Applications*, p. 87–94, Amsterdam, The Netherlands, août 1994.

[JÉG 95] JÉGOU P. et BOUQUET F., Solving over-constrained CSP using weighted OBDD. *In : Proceedings of the CP'95 Workshop on Over-Constrained Systems*, Cassis, France, septembre 1995.

[SCH 95] SCHIEX T., FARGIER H. et VERFAILLIE G., Valued constraint satisfaction problems: hard and easy problems. *In : Proc. of the 14th IJCAI*, p. 631–637, Montréal, Canada, août 1995.

[SCH re] SCHIEX T., FARGIER H. et VERFAILLIE G., Problèmes de satisfaction de contraintes valués. *Revue d'Intelligence Artificielle*, 1996, à paraiître.

[VER 95] VERFAILLIE G. et SCHIEX T., Maintien de solution dans les problèmes dynamiques de satisfaction de contraintes : bilan de quelques approches. *Revue d'Intelligence Artificielle*, 1995.