QCSP$^+$non-bloquants : un cas spécial de problèmes quantifiés

Arnaud Lallouet et Jérémie Vautard

Université d'Orléans — LIFO
BP 6759 — F-45067 Orléans cedex 2
jeremie.vautard@univ-orleans.fr

Résumé. Ce papier présente un cas spécial de QCSP$^+$appelé QCSP$^+$non-bloquants, dans lesquels la les restrictions posées sur les quantificateurs ne vident jamais le domaine d'une variable. Intuitivement, ces cas spéciaux correspondent à des jeux opposant deux adversaires dans lesquels il n'est pas possible de bloquer toute possibilité de mouvement d'un joueur. Nous présentons des techniques de résolution basées sur ce cas spécial, ainsi que des exemples de modèles non-bloquants.

1 Introduction

Le formalisme de la programmation par contraintes quantifiées (QCSP) étend les CSP classiques en ajoutant des quantificateurs universels ou existentiels sur les variables, ce qui permet de modéliser des problèmes de complexité supérieure à NP, comme décider si, dans un jeu à deux joueurs, l'un des joueurs a la possibilité de jouer de manière à toujours gagner, quoi que puisse faire son adversaire.

La solution d'un tel problème ne peut pas être une simple affectation de toutes ses variables : en effet, les contraintes doivent être satisfaites quelles que soient les valeurs affectées aux variables quantifiées universellement. On peut voir une telle solution comme une famille de fonctions de Skolem, chacune de ces fonctions affectant à une variable existentielle une valeur de son domaine en fonction des valeurs des variables universelles qui la précède. On appelle une telle famille de fonctions une *stratégie*. Lorsque cette famille est telle que, pour toutes les affectations possibles des variables universelles, l'affectation des variables existentielles qui en découle satisfait toutes les contraintes du problèmes, il s'agit d'une *stratégie gagnante*. Un QCSP est vrai si, et seulement si il admet une telle stratégie gagnante.

Le terme *stratégie* vient du fait qu'un QCSP peut être vu comme un jeu opposant le "joueur universel" au "joueur existentiel", consistant à affecter tour à tour les variables du problème, le joueur existentiel cherchant à résoudre toutes les contraintes du problème tandis que le joueur universel cherche à l'en empêcher. De ce point de vue, les fonctions de Skolem constituant une stratégie gagnante indiquent au joueur existentiel les coups à jouer de manière à ce qu'il soit sûr de gagner.

Cette ressemblance a conduit [1] à définir une extension à ce formalisme, appelée QCSP$^+$. Cette extension rend la modélisation de problèmes plus aisée en permettant de décrire directement sous la forme d'un CSP les règles du jeu, en restreignant la portée des quantificateurs universels aux seules solutions de ce CSP. Ceci permet même à ce formalisme de décrire des cas où l'un des joueurs "perd la partie" faute de coup possible à jouer. Cependant, pour les instances où l'on sait que ce dernier cas ne peut pas se produire, il est possible d'améliorer l'algorithme de résolution décrit dans [1]. Cet article présente ces améliorations.

2 QCSP

Notations. Soit V un ensemble de variables, D étant la famille de leurs domaines. On note D_X le domaine d'une variable X. Soit W un sous-ensemble de V. On note D^W l'ensemble des n-uplets sur W, c'est à dire le produit cartésien $\Pi_{X \in W} D_X$ des domaines des variables de W. Par ailleurs, on note $|$ la projection d'un (ensemble de) tuples sur une ou plusieurs variables.

Contraintes et CSP. Une *contrainte* $c = (W, T)$ est composée d'un sous-ensemble de variables $W \subseteq V$ et d'une relation $T \subseteq D^W$. W et T sont aussi respectivement notés $var(c)$ e $sol(c)$. Une contrainte vide, c'est-à-dire telle que $sol(c) = \emptyset$ est *fausse* et notée $\perp$, tandis qu'une contrainte est dite *pleine* et notée $\top$ si, et seulement si $sol(()c) = D^W$.

Un *problème de satisfaction de contraintes*, ou *CSP*, est un ensemble de contraintes. On note $var(C) = \bigcup_{c \in C} var(c)$ l'ensemble de ses variables et $sol(C) = \bowtie_{c \in C} sol(c)$ l'ensemble de ses solutions, i.e. l'ensemble de toutes les affectations des variables de $var(C)$ qui satisfont toutes les contraintes. Un CSP vide est *vrai* et sera noté $\top$ tandis qu'un CSP contenant une contrainte vide est lui-meme trivialement faux et sera noté $\perp$.

Contraintes quantifiées et QCSP. On appelle *qset* un couple (q, W) où $q \in \{\exists, \forall\}$ est un quantificateur et $W \subseteq V$ un sous-ensemble de variables.

Définition 1 (Préfixe). *Un* préfixe P *est une séquence de qsets* $[(q_0, W_0), \ldots, (q_{n-1}, W_{n-1})]$ *dans laquelle* $i \neq j \Rightarrow W_i \cap W_j = \emptyset$.

On note $P|_W$ la restriction de P aux variables de l'ensemble W.

On dit qu'une variable X est *déclarée* dans un qset (q_i, W_i) si $X \in W_i$. Un QCSP est alors défini en ajoutant un CSP à un préfixe ;

Définition 2 (QCSP). *Un* CSP quantifié *ou* QCSP *est un couple* (P, G) *dans lequel* P *est un préfixe et* G *un CSP appelé* goal *tel que* $var(G) \subseteq var(P)$.

Exemple 3 (QCSP). La formule suivante :

$$\exists X \in \{0,1\},\ \forall Y \in \{0,1\},\ \exists Z \in \{1,2\}\ .\ X + Y = Z$$

est représentée par le QCSP suivant :

$$Q = ([(\exists, X), (\forall, Y), (\exists, Z)], \{X + Y = Z\})$$

Dans cet exemple, le préfixe est $[(\exists, X), (\forall, Y), (\exists, Z)]$ tandis que le goal se limite à l'unique contrainte $\{X + Y = Z\}$

Résolution Comme nous l'évoquions en introduction, contrairement au cas des CSP, la notion de solution d'un QCSP ne peut plus être une simple affectation des variables du problème. Une solution doit en effet expliciter les valeurs prises par les variables existentielles en fonction des variables universelles de manière à ce que le goal reste satisfait quelles que soient les valeurs prises par ces dernières. On appelle une telle solution une *stratégie gagnante*, qui peut être vue comme un ensemble de fonctions de Skolem, donnant une valeur à chaque variable existentielle en fonction des variables universelles précédentes (comme c'est le cas par exemple dans [2]). Le fait qu'une telle stratégie gagnante existe pour un QCSP donné prouve la formule logique associée à ce QCSP. On note $\textsc{win}(Q)$ l'ensemble des stratégies gagnantes d'un QCSP Q.

Sémantique Nous utilisons dans ce papier une sémantique purement *décisionnelle* des QCSP, c'est-à-dire donnant une valeur $\top$ ou $\perp$ à un QCSP.

Définition 4 (Sémantique décisionnelle d'un QCSP). *La sémantique décisionnelle $[\![Q]\!]|_d$ d'un QCSP Q est :*

$[\![Q]\!]|_d = \top$ si $Win(Q) \neq \emptyset$, $\perp$ sinon.

$QCSP^+$. Les $QCSP^+$ introduisent des restricteurs sur les quantificateurs d'un QCSP en modifiant la nature du préfixe : en plus du quantificateur et d'un ensemble de variables, chaque qset inclut un CSP dont les solutions sont les valeurs "autorisées" pour ses variables. Les $QCSP^+$ ont été présentés pour la première fois dans [1] et motivés principalement par des problèmes de modélisation en QCSP classiques.

On définit un *rqset* comme un triplet (q, W, C) où (q, W) contitue un qset tel que définit plus haut, et C un CSP que l'on appelle *restricteur*. Le sens d'un tel objet est de restreindre les variables de W à prendre les seuls valeurs satisfaisant C. On étend la notion de préfixe aux rqsets en continuant à requérir la condition $i \neq j \Rightarrow W_i \cap W_j = \emptyset$. De plus, chaque CSP C_i doit être tel que $var(C_i) \subseteq (W_1 \cup \ldots \cup W_i)$.

Définition 5 ($QCSP^+$). *Un $QCSP^+$ est un couple $Q = (P, G)$ avec P un préfixe de rqsets et G le CSP goal.*

Notons qu'un QCSP classique peut être vu comme un $QCSP^+$dont tous les C_i sont vides. La définition de la stratégie d'un $QCSP^+$est la même que celle d'un QCSP. en revanche, celle de stratégie gagnante change légèrement : une stratégie gagnante est une stratégie pour laquelle tout coup possible du joueur universel se termine par un scénario gagnant. Comme dans un QCSP classique, cela peut se produire quand toutes les contraintes du goal sont satisfaites, mais il peut aussi arriver que le CSP d'un rqset universel devienne inconsistant, auquel cas le scénario devient vrai, quelle que soit l'affectation des variables restantes. La sémantique d'un QCSP$^+$ peut donc être définie de manière inductive :

Définition 6 (Sémantique d'un QCSP$^+$). *La sémantique $[\![Q]\!]$ d'un QCSP$^+$ Q est définie de la manière suivante :*

- $[\![([], G)]\!] = (sol(G) \neq \emptyset \ ? \ \top : \bot)$.
- $[\![([(\exists, W, C)|P'], G)]\!] = \bigvee_{t \in sol(C)}([\![P'[W \leftarrow t], G[W \leftarrow t]]\!])$
- $[\![([(\forall, W, C)|P'], G)]\!] = \bigwedge_{t \in sol(C)}([\![P'[W \leftarrow t], G[W \leftarrow t]]\!])$

Comme nous l'avons évoqué plus haut, $sol(C)$ peut être vide. Dans ces cas, on considère que $\bigvee(\emptyset) = \bot$ et $\bigwedge(\emptyset) = \top$.

3 QCSP$^+$ non-bloquants

3.1 Intuition et définition

Soit Q un QCSP$^+$ (P, G) tel que $P = [(q_0, W_0, C_0), \ldots, (q_{n-1}, W_{n-1}, C_{n-1}]$. Considérons-le sous l'angle du jeu entre le joueur $\exists$ et le joueur $\forall$. Le premier joueur affecte des valeurs aux variables W_0 de manière à satisfaire C_0. Ensuite, son adversaire affecte W_1 tel que C_1 soit satisfait, et ainsi de suite jusqu'à ce que toutes les variables soient instanciées. Dès lors, le joueur $\exists$ gagne si le goal est satisfait, son adversaire remportant la partie dans le cas contraire.

Dans le cas général, il est possible que ce jeu se termine "prématurément", du fait que l'un ces CSP C_i devient inconsistant. Par exemple, si le préfixe P contient dès le début un rqset de la forme $(q_i, W_i, \bot)$, alors le gagnant sera toujours connu avant que l'on atteigne la fin du jeu : si on en vient à atteindre la profondeur i, le joueur correspondant n'a pas de mouvement valide (il est impossible d'affecter les variables de W_i demanière à satisfaire C_i), et perd donc instantanément la partie.

Ceci est un exemple (trivial) de ce que nous appelons un préfixe *bloquant* : dans au moins un scénario, l'un des joueur perd le jeu, non pas en fonction de la valeur de vérité du goal, mais parce qu'il a été empêché de jouer à un moment donné du jeu. Notons que dans le cas général, seul quelques scénarios sont bloquants dans un préfixe. Dans le cas contraire, on dit que le préfixe est *non-bloquant*, si et seulement si tous les scénarios possibles selon le préfixe atteignent le goal.

Formellement, cette propriété signifie qu'à tout niveau i du préfixe, (1) il existe une affectation A des variables de $W_0 \cup \ldots \cup W_{i-1}$ consistante avec l'ensemble des CSP $C_0 \cup \ldots \cup C_{i-1}$, et (2) pour toutes les affectations répondant à cette définition, il existe au moins une affectation A_i des variables de W_i telle que $A \cup A_i$ est une solution de C_i. Notons que cette propriété porte uniquement sur le *préfixe* du problème. Nous définissons donc cette propriété comme étant une propriété du préfixe d'un QCSP$^+$.

Définition 7 (préfixe non-bloquant).

Soit $P = [(q_0, W_0, C_0), \ldots, (q_{n-1}, W_{n-1}, C_{n-1}]$ *un préfixe de longueur* n. P *est* non-bloquant *si et seulement si : (1) il s'agit du préfixe vide OU (2)* C_0 *admet au moins une solution, et, pour toute solution* s *de* C_0*, le préfixe* $P|_a$ *défini comme étant* $[(q_1, W_1, C_1[W_0 \leftarrow a]),\ \ldots,\ (q_{n-1}, W_{n-1}, C_{n-1}[W_0 \leftarrow a])]$ *est lui-même non-bloquant.*

On étend cette propriété au QCSP lui-même en transposant celle de son préfixe :

Définition 8 (QCSP$^+$ non-bloquant). *Un QCSP*$^+$ $Q = (P, G)$ *est non-bloquant si et seulement si* P *l'est.*

Notons que les QCSP classiques sont des exemples triviaux de QCSP$^+$ non-bloquants.

Exemple : le jeu de Nim. Le jeu de Nim consiste en un tas d'allumettes duquel deux joueurs retirent tour à tour entre un et trois éléments. Le joueur qui vide le tas gagne la partie. On peut modéliser ce jeu en QCSP$^+$ de différentes manières, certaines donnant des QCSP$^+$ non-bloquants. Considérons par exemple un tas initial de 20 allumettes. Voici un premier modèle du jeu :

$$
\begin{array}{l}
\exists X_1 \in [1..3][] \\
\ \forall Y_1 \in [1..3], S_1^{\forall} \in [0..20][S_1^{\forall} = X_1 + Y_1] \\
\ \ \exists X_2 \in [1..3], S_2^{\exists} \in [0..20][S_2^{\exists} = X_2 + S_1^{\forall} \wedge S_1^{\forall} \neq 20] \\
\ \ \ \vdots \\
\ \ \ \exists X_i \in [1..3], S_i^{\exists} \in [0..20][S_i^{\exists} = X_i + S_{i-1}^{\forall} \wedge S_{i-1}^{\forall} \neq 20] \\
\ \ \ \forall Y_{i+1} \in [1..3], S_{i+1}^{\forall} \in [0..20][S_{i+1}^{\forall} = Y_{i+1} + S_i^{\exists} \wedge S_i^{\exists} \neq 20] \\
\ \ \ \ \vdots \\
\ \ \ \ \top
\end{array}
$$

Dans ce modèle, les variables $S_i^{\exists}$ et $S_i^{\forall}$ représentent le nombre d'allumettes retirées depuis le début de la partie au tour i. La condition de victoire est directement modélisée dans les rqsets par les contraintes $S_i^{\exists} \neq 20$. En effet, on viole une telle contrainte à partir du moment où l'adversaire s'est emparé de la dernière allumette du tas. Ainsi, dans ce cas, on rend le rsqet suivant inconsistant, de sorte que l'adversaire perde la partie. Ce modèle est clairement bloquant, étant donné qu'un joueur est empêché de jouer dès que son adversaire s'est emparé de la dernière allumette. De fait, aucun scénario n'atteindra jamais le goal, étant donné que la dernière allumette sera toujours prise.

Proposons un modèle non-bloquant pour ce jeu. Dans celui-ci, nous explicitons les conditions de victoire du joueur $\exists$ directement dans le goal : la modélisation du décompte des allumettes reste dans les rqsets, mais ceux-ci ne bloqueront plus l'adversaire du joueur qui retire la dernière allumette. Le goal se voit en effet augmenté des contraintes assurant que le joueur $\exists$ a bien pris la dernière allumette :

$\exists X_1 \in [1..3][]$
$\forall Y_1 \in [1..3], S_1^{\forall} \in [0..60][S_1^{\forall} = X_1 + Y_1]$
$\exists X_2 \in [1..3], S_2^{\exists} \in [0..60][S_2^{\exists} = X_2 + S_1^{\forall}]$
$\vdots$
$\exists X_i \in [1..3], S_i^{\exists} \in [0..60][S_i^{\exists} = X_i + S_{i-1}^{\forall}]$
$\forall Y_{i+1} \in [1..3], S_{i+1}^{\forall} \in [0..60][S_{i+1}^{\forall} = Y_{i+1} + S_i^{\exists}]$
$\vdots$
$[S_2^{\exists} = 20 \vee S_3^{\exists} = 20 \vee S_4^{\exists} = 20 \vee \ldots$

Ici, les domaines des variables S restent assez grands afin de ne jamais pouvoir se vider (cette taille n'est cependant pas un problème, la valeur de ces variables étant fixées dès que les autres variables sont affectées). Le goal est vrai si et seulement si le nombre d'allumettes retirées par le joueur $\exists$ est égal au nombre initial d'allumettes présentes dans le tas en début de partie, c'est à dire si ce joueur vide le tas à un moment ou à un autre. Dans le cas ou son adversaire vide le tas, cette contrainte ne peut plus être satisfaite, et le goal devient inconsistant.

Ce modèle peut aisément être reconnu comme étnant non-bloquant : en effet, les CSP C_i se contentent de lier les valeurs des sommes partielles S aux nombres d'allumettes retirées à chaque tour X et Y et ne peuvent en aucun cas devenir totalement inconsistants.

3.2 Propagation dans les QCSP$^+$ non-bloquants

Le fait qu'un QCSP$^+$ soit non-bloquant a deux conséquences : d'une part, nous savons qu'une affectation des variables W_i qui rendrait inconsistant un CSP C_j situé plus loin dans le préfixe ne sera pas consistante non plus avec le CSP C_i. D'autre part, si à un moment donné, le goal devient inconsistant ou trivialement vrai, alors étant donné que tous les scénarios arrivent toujours au goal, le problème prend automatiquement la valeur de vérité du goal.

La formalisation de ces deux points donne lieu à une procédure de propagation additionnelle pour les QCSP$^+$ non bloquants : la première conduit à une fusion de tous les ensembles de contraintes présent dans les rqsets, et la seconde à une vérification du goal à chaque affectation de l'ensemble de variables d'un rqset.

Fusion des restricteurs. Du point de vue de la recherche de solution, le premier point évoqué nous permet de considérer n'importe quelle contrainte de n'importe quel rqset comme faisant aussi partie du restricteur courant. La propagation peut ainsi tirer parti de toutes ces contraintes lors de la recherche d'une affectation valide. Pour prouver la correction de cette technique, il nous faut montrer que toute valeur x d'une variable X de W_0 qui serait inconsistante dans un restricteur C_i l'est aussi dans C_0. Étant donné la définition inductive de la propriété d'être non-bloquant, cette démonstration s'étendra à toutes les variables du problème.

Théorème 9 (Fusion des restricteurs). *Soit $Q = (P, G)$ un QCSP$^+$ non-bloquant tel que $P = [(q_0, W_0, C_0), \ldots, (q_n, W_n, C_n)]$, et X une variable de son* *premier rqset ($X \in W_0$). Toute valeur x de D_X qui est inconsistante avec un des restricteurs C_i tel que $1 \leq i \leq n$ est inconistante avec C_0.*

Preuve. Q étant non-bloquant, C_0 possède au moins une solution. Supposons qu'il existe une valeur x de D_X consistante avec C_0, mais pas avec un certain C_k. Soit S_0 une solution de C_0 affectant x à X.

Hypothèse d'nduction : Si un restricteur C_i avec $0 < i < k$ admet une solution S_k affetant x à X, et telle que $S_i|W_0 \in sol(C_0), \ldots, S_i|(W_0 \cup .. \cup W_{i-1} \in sol(C_{i-1})$, alors, comme P est non-bloquant, il existe donc une solution S_{i+1} de C_{i+1} telle que $S_{i+1}|(W_0 \cup .. \cup W_k) = S_i$ (et donc, où $X = x$).

Donc, par induction, C_k admet une solution S_k affectant la valeur x) X, ce qui contredit l'hypothèse selon laquelle x est inconsistante pour C_k. Par conséquent, si l'affectation $X \leftarrow x$ est inconsistante dans C_k, elle est aussi inconsistante dans C_0.

Vérification du goal. Comme nous l'avons vu plus haut, le fait qu'un QCSP$^+$ soit non-bloquant implique que si une inconsistance est détectée dans le goal (e.g. par propagation), alors le problème entier peut être évalué à $\perp$. De manière similaire, si le goal devient trivialement vrai avec les domaines de variables atuels, alors le QCSP$^+$ entier peut être directement évalué) $\top$:

Théorème 10 (Vérification du goal). *Soit $Q = (P, G)$ un QCSP$^+$ non bloquant, tel que $P =$ $[(q_0, W_0, C_0), \ldots, (q_{n-1}, W_{n-1}, C_{n-1}]$. Si $sol(G) = \emptyset$, alors $[\![Q]\!] = \perp$, et si $sol(G) = D^{W_0 \cup \ldots \cup W_{n-1}}$, alors $[\![Q]\!] = \top$.*

Preuve. Si $sol(G) = \emptyset$, par définition, on a $[\![([], G)]\!] = \perp$. Considérons un préfixe non-bloquant $P' = [(q, W, C)|P'']$ et supposons que pour toute solution S de C, $[\![(]\!]P'', G[C \leftarrow S]) = \perp$ (

La déinition de la sémantique d'un QCSP$^+$ nous donne, selon que $q = \exists$ ou que $q = \forall$, soit $[\![(]\!]P', G) = \bigvee_{t \in sol(C_i)}([\![P''[W \leftarrow t], G[W \leftarrow t])]\!]$, soit $[\![(]\!]P', G) = \bigwedge_{t \in sol(C_i)}([\![(P''[W \leftarrow t], G[W \leftarrow t])]\!])$. Mais, P' étant lui-même un préfixe non)bloquant, $sol(C)$ n'est pas vide. Par l'hypothèse d'induction, on a donc soit $[\![(P', G)]\!] = \bigvee_{t \in sol(C_i)}(\perp)$ ou $[\![(P', G)]\!] = \bigwedge_{t \in sol(C_i)}(\perp)$ ce qui, dans les deux cas, nous mène à $[\![(P', G)]\!] = \perp$.

La démonstration de la seconde partie est analogue.

3.3 QCSP$^+$semi-bloquants

La procédure de résolution d'un QCSP$^+$ passe le plus clair de son temps à rechercher une solution pour un rqset. Cette recherche de solution peut bénéficier de la technique de fusion des restricteurs, mais la vérification du goal ne peut se faire que lorsque l'ensemble des variables d'un qset vient d'être affecté. En effet, il est impossible, en plein milieu de la résolution d'un restricteur, de savoir si l'état de recherche courant va aboutir à une solution ou non. On ne peut donc pas utiliser la vérification du goal en prenant en compte les réductions de domaines faites sur ce restricteur, étant donné que rien ne dit si ces réductions vont effectivement aboutir à une solution ou non

Considérons le cas d'un QCSP$^+$ $Q = ([(q_0, W_0, C_0)|P'], G)$. Si ce problème à tout point de la recherche de solution du premier restricteur C_0, il n'y a que deux possibilités :

- soit l'affectation partielle de W_0effectuée par la recherche de cette solution ne rend pas ce dernier inconsistant, et donc Q est toujours non-bloquant ;
- soit C_0 est devenu inconsistant (mais on ne l'a pas encore détecté), et la prise en compte de l'affectation partielle des variables de W_0 rend donc Q vrai si $Q_0 = \forall$, ou faux si $Q_0 = \exists$.

Dans les deux cas, notons que si $q_0 = \exists$ et $G = \bot$ (resp. $q_0 = \forall$ et $G = \top$), on a $[\![Q]\!] = \bot$ (resp.$[\![Q]\!] = \top$).

Formalisons ce cas spécial :

Définition 11 (Préfixe semi-bloquant).
soit $P = [(q_0, W_0, C_0), \ldots, (q_{n-1}, W_{n-1}, C_{n-1}]$ un préfixe de longueur n. P est ∃-semi-bloquant *(resp.* ∀-semi-bloquant *si, et seulement si : (1) $q_0 = \exists$ (resp. $q_0 = \forall$) et (2) pour toute solution S de C_0, le préfixe $P|_S = [(q_1, W_1, C_1[W_0 \leftarrow S]), \ldots, (q_{n-1}, W_{n-1}, C_{n-1}[W_0 \leftarrow S])]$ est non-bloquant.*

Définition 12 (QCSP$^+$ semi-bloquant).
Un QCSP$^+$ $Q = (P, G)$ est ∃-semi-bloquant *(resp.* ∀-semi-bloquant*) si, et seulement si P is* ∃-semi-bloquant *(resp.* ∀-semi-bloquant*).*

La différence entre cette définition et la définition de préfixe / QCSP$^+$ non-bloquant est que l'existence d'une solution pour C_0 n'est plus requise. Cependant, Si il en existe, alors le problème est bien non-bloquant. Dans chacun de ces cas, on peut appliquer une "moitié" du théorème de vérification du goal :

Théorème 13 (∃-semi vérification du goal).
Soit $Q = (P, G)$ un QCSP$^+$ ∃-semi-bloquant, avec $P = [(\exists, W_0, C_0), \ldots, (q_{n-1}, W_{n-1}, C_{n-1}]$. Si $sol(G) = \emptyset$, alors $[\![Q]\!] = \bot$.

Preuve. Si $sol(C_0) = \emptyset$, alors par définition de la sémantique des QCSP$^+$, $[\![Q]\!] = \bot$. Sinon, Q est non-bloquant et le théorème de vérification du goal s'applique.

Théorème 14 (∀-semi vérification du goal).
Soit $Q = (P, G)$ un QCSP$^+$ ∀-semi-bloquant, tel que $P = [(\forall, W_0, C_0), \ldots, (q_{n-1}, W_{n-1}, C_{n-1}]$. Si $sol(G) = \prod(D(W_0) \ldots D(W_{n-1})$, alors $[\![Q]\!] = \top$.

Preuve. Si $sol(C_0) = \emptyset$, $[\![Q]\!] = \top$. Sinon, Q est non-bloquant et le théorème de vérification du goal s'applique également.

4 Discussion

Une propriété à indiquer a-priori. Intuitvement, un QCSP$^+$ est non-bloquant si, et seulement si on ne rencontre jamais pendant la résolution de restricteur sans solution. Du point de vue du jeu opposant le joueur-$\exists$ au joueur-$\forall$, tout joueur a au moins un coup valide à jouer jusqu'à la fin de la partie. En d'autres termes, il existe un mouvement possible pour le joueur actuel, et pour tous ces mouvements possibles, son adversaire aura un mouvement possible, et pour tous ses mouvements, etc.

Donc, décider si un préfixe $P = [(q_0, W_0, C_0), \ldots, (q_{n-1}, W_{n-1}, C_{n-1})]$ est non-bloquantrevient à décider la valeur de vérité de la formule suivante :

$$\exists w_0 \in D_{W_0} C_0 \wedge \forall w'_0 \in D_{W_0} C_0 \rightarrow \exists w_1 \in D_{W_1} C_1 \wedge \forall w'_1 \in D_{W_1} \ldots$$

ce qui est équivalent à résoudre le QCSP$^+$ suivant :

$\exists W_0[C_0(W_0)]$
 $\forall W'_0[C_0(W'_0)]$
  $\exists W_1[C_1(W'_0, W_1]$
   $\forall W'_1[C'_1(W'_0, W'_1)]$
    $\ddots$
     $\top$

Décider si un QCSP$^+$ donné est non-bloquant est donc en général aussi complexe que de le résoudre. Ainsi, utiliser les techniques présentées dans ce papier n'a de sens que si le modeleur fournit un QCSP$^+$ construit de manière à être non-bloquant et le déclare comme tel.

Cependant, on pourrait détecter certaines instances comme “aisément non-bloquantes”. Par exemple, on peut imaginer un QCSP$^+$ où les restricteurs portent sur des ensembles de variables disjoints deux à deux. Dans ce cas, il suffit, pour assurer qu'un tel QCSP$^+$ est non-bloquant, de vérifier que tous les restricteurs ont au moins une solution.

Planification avec adversaire “non-bloquante”. Les QCSP$^+$ non-bloquants ne se limitent pas à la modélisation de jeux. [3] présente un petit modèle de planification avec adversaire, où un ordonnancement de tâches doit rester faisable quoi que puisse faire un ennemi. Ce modèle est du type :

$\exists(S)[Faisable(S)]$
 $\forall(A)[Possible(A, S)]$
  $Faisable(A(S))$

où S est un ordonnancement (*schedule*), A la modélisation d'une attaque de l'ennemi, $A(S)$ les données de l'ordonnancement après cette attaque, $Faisable$ un ensemble de contraintes assurant qu'un ordonnancement est faisable, et $Possible$ un ensemble de contraintes déterminant si une attaque est dans les possibilités de l'ennemi ou non.

Dans ces problèmes, il est toujours possible de trouver un ordonnancement réalisable avant passage de l'ennemi, et celui-ci peut toujours déclencher une attaque. Ces problèmes sont donc non-bloquants, et les techniques présentées ici peuvent donc s'appliquer.

Problèmes partiellement non-bloquants. Beaucoup de problèmes, notamment des jeux, sont naturellement modélisés de manière à ce que les premiers restricteurs aient toujours des solutions. Par exemple, considérons le premier modèle du jeu de Nim présenté en section 3. Ce modèle est clairement bloquant. Cependant, étant donné que chaque joueur peut prendre au maximum trois allumettes, la vingtième et dernière allumette ne peut pas être prise avant le septième tour. On peut imagine un extension de la propriété des QCSP$^+$ non-bloquants qui définirait ce modèle comme *non-bloquant jusqu'au* 7^{eme} *rqset*, ce qui permettrait de fusionner les sept premiers restricteurs.

Modélisation. Dans [4], Peter Nightingale présente des modèles en QCSP standatrd pour le jeu du morpion et celui du Puissance-4. Dans ces modèles, il créé des "clones" existentiels des variables universelles, et pose des contraintes *shadow* assurant qu'une variable et son clone doivent être égales si le coup joué est valide. Dans le cas contraire (i.e. si le joueur universel "triche"), le clone existentiel peut être fixé à n'importe quelle valeur valide, ce qui correspond intuitivement au fait que le joueur existentiel joue à la place de son adversaire, et peut donc effectuer un mouvement qui soit en sa faveur. Cependant, cette technique ne peut pas s'appliquer dans les cas où le joueur-$\forall$ n'a plus aucun coup valide à jouer. en effet, dans ce cas, le clone existentiel de la variable universelle voit aussi son domaine se vider par les contraintes modélisant les règles du jeu. Le QCSP s'évalue donc dans ce cas à $\perp$ alors qu'il devrait au contraire être vrai. Ainsi, cette technique de modélisation impose que le joueur-$\forall$ aie toujours un coup valide à jouer, i.e. qu'il soit soumis à des règles non-bloquantes.

On remarque donc une certaine corrélation entre les problèmes pouvant naturellement se modéliser par un QCSP$^+$ non-bloquant sont ceux qui peuvent se modéliser aisément en QCSP standard grâce à cette technique présentée par Nightingale. Cependant il existe aussi, comme pour le jeu de Nim présenté plus haut, des modèles bloquants pour ces problèmes. Il reste donc à étudier les différents couples (modèle,technique de résolution) pour différents problèmes, afin de comparer ces approche entre elles sur une gamme de problèmes large.

5 Conclusion

Ce papier a exhibé le cas spécial des QCSP$^+$ non-bloquants, où des techniques de résoution additionnelles peuvent être appliquées. Ce cas spécial correspond naturellement à toute une classe de jeux où aucun joueur ne peut être empêché de jouer, de même qu'a d'autres problèmes moins ludiques, comme certains problèmes d'ordonnancement avec adversaire. Savoir a-priori qu'un QCSP$^+$ est non-bloquant (e.g. par construction) est susceptible d'accélérer sensiblement sa résolution.

Les travaux présentés dans ce papier sont préliminaires à une étude plus large des problèmes quantifiés pouvant aisément se modéliser de manière non-bloquante, afin de situer les différentes techniques de modélisation les unes par rapport aux autres.

References

1. Benedetti, M., Lallouet, A., Vautard, J.: QCSP Made Practical by Virtue of Restricted Quantification. In Veloso, M., ed.: International Joint Conference on Artificial Intelligence, Hyderabad, India, AAAI Press (2007) 38–43
2. Bordeaux, L., Cadoli, M., Mancini, T.: CSP properties for quantified constraints: Definitions and complexity. In Veloso, M.M., Kambhampati, S., eds.: National Conference on Artificial Intelligence, AAAI Press (2005) 360–365
3. Benedetti, M., Lallouet, A., Vautard, J.: Modeling adversary scheduling with QCSP+. In: ACM Symposium on Applied Computing, Fortaleza, Brazil, ACM Press (2008)
4. Nightingale, P.: Consistency and the Quantified Constraint Satisfaction Problem. PhD thesis, University of St Andrews (2007)