Noyau de viabilité : une contrainte globale pour la modélisation de systèmes dynamiques

Juliette Mattioli, Konstantin Artiouchine

Noyau de viabilité : une contrainte globale pour la modélisation de systèmes dynamiques

Juliette Mattioli* — **Konstantin Artiouchine*,****

* *Thales TRT, Domaine de Corbeville, 91404 Orsay CEDEX,France*
{Konstantin.Artiouchine, Juliette.Mattioli}@thalesgroup.com

** *LIX, Ecole Polytechnique, 91128 Palaiseau CEDEX, France*
Konstantin.Artiouchine@polytechnique.org

RÉSUMÉ. *L'objectif de ce papier est de décrire une approche pour aborder la problématique induite par la nature hybride (statique + dynamique) des systèmes. En effet, étant donné un système dynamique, discret ou continu, et un ensemble K défini par un système de contraintes indépendantes du temps, il n'est pas certain qu'à partir d'un point initial x_0, il existe une solution du système dynamique qui reste toujours dans K. On s'intéresse alors à l'ensemble des valeurs initiales de K à partir lesquels est issu au moins une solution qui ne sort jamais de K. Le plus grand sous-ensemble obtenu est appelé le "noyau de viabilité" de K et la transformation de K en $Viab(K)$ est en fait un opérateur de propagation. Grâce à l'introduction de cette notion au travers d'une contrainte globale dans un solveur de contraintes en domaine fini, nous pouvons aborder certains problèmes hybrides définis par la combination d'un système de contraintes (classiques) et d'un système dynamique.*

ABSTRACT. *This paper presents an approach to handle some problems arising in the modelling of hybrid systems combining static and dynamic. In the case of a discrete or continuous dynamic system, there is no particular condition forcing a solution of a differential inclusion starting from a point x_0 to remain in a set K defined by a set of static constraints. We study the subsets of K containing only the points such that at least one solution starting from them will stay in K forever. The largest subset of K containing all such points is called the "viability kernel" of K, and the operator transforming K into $Viab(K)$ is a propagation operator. This allows the notion of viability to be introduced as a global constraint which enables us to model certain hybrid problems defined as a combination of classic constraints with a dynamic system.*

MOTS-CLÉS : *Programmation par contraintes, système hybride, théorie de la viabilité.*

KEYWORDS: *Constraint programming, hybrid system, viability theory.*

1. Motivation de l'approche

Pour pouvoir prendre une décision dans un contexte évolutif, tout en gérant globalement leurs ressources, les systèmes d'informations doivent avoir une représentation synthétique mais précise de la réalité. Que la décision se fasse en boucle fermée sur l'environnement, afin d'être réactive aux changements, ou qu'elle soit en boucle ouverte, il est crucial que la conception s'appuie sur une modélisation réaliste (et donc complexe) de l'environnement ou du contexte. Elle doit aussi aider le décideur en proposant des choix optimaux d'affectation de tâches et de ressources, et ceci dans les meilleurs délais. Il s'agit donc de raisonner sur l'état présent et sur une estimation future d'un environnement changeant. Pour illustrer notre propos, citons les fonctions intelligentes de gestion multi-capteurs, les fonctions de choix automatiques de paramètres de réglages en fonction du contexte, utiles pour les équipements ou les fonctions de supervision (contrôle aérien, surveillance de zone, ...). Ces problématiques auxquelles s'intéresse Thales, ont leurs ressources en nombre fini, et l'environnement dans lequel elles évoluent est dynamique.

Ces exemples sont non seulement des problèmes combinatoires, mais sont ainsi régis par un système dynamique défini par des équations différentielles ou des inclusions différentielles suivant le degré de précision du modèle que l'on a. Ces problèmes intègrent donc plusieurs aspects : la description précise du contexte initial et des contraintes du problème (liées par exemple à la gestion et à la disponibilité des ressources), la description de la dynamique et des états du système, ainsi que l'objectif final de la décision avec éventuellement les différentes heuristiques métiers permettant d'orienter les décisions.

La modélisation et la résolution de ces systèmes ne se cantonnent alors pas à une approche bien définie, mais intègrent plusieurs sous-problèmes combinant la complexité liée à la nature combinatoire des contraintes et à la nature dynamique du système. Par ailleurs, le paradigme sous jacent à la PPC permet au développeur de ne pas se soucier de l'ordre dans lequel les contraintes sont introduites. La conséquence principale est qu'elle rend possible une programmation incrémentale. Cela signifie que si l'on désire modifier ou ajouter un nouveau modèle, on peut le faire sans avoir à se soucier des interactions avec les modèles existants.

L'objectif de cet article est de présenter une approche prise par Thales pour aborder la problématique induite par la nature hybride (statique+dynamique) de ces systèmes. L'idée principale est d'enrichir un solveur de PPC à l'aide de certains résultats de la théorie de la "viabilité" [AUB 91] permettant ainsi d'aborder la conception d'un système dynamique d'aide à la décision, tout en prenant en compte l'hétérogénéité sous jacente aux aspects statiques/dynamiques. De manière générique, un tel système peut se composer de définitions de variables sur lesquelles reposent les contraintes du système (ressources, contraintes d'utilisation, ...) mais aussi par un système dynamique caractérisant l'évolution des états du système.

2. Une brève présentation de la théorie de la viabilité

Depuis la fin des années 80 s'est développée à partir de l'analyse multivoque [AUB 84, AUB 90], la théorie de la viabilité [AUB 91] dont la principale motivation était l'étude des évolutions des systèmes complexes en avenir incertain soumis à des contraintes de viabilité (d'où le nom). Son champ d'application, initialement centré sur les systèmes évolutionnaires en sciences économiques, s'est considérablement élargi dans les années 90 aux sciences sociales (économie dynamique [AUB 97], démographie [BON 97], finance [DOY 96], environnement [DOY 96, DOY 97], ...) et aux sciences de l'ingénieur (contrôle [QUI 91, SAI 98], jeux différentiels[CAR 98], optimisation continue [GOR 96, AUB 98], optimisation de formes [AUB 99]). Cette théorie offre actuellement un cadre mathématique adapté à l'étude des problèmes de contrôle dynamique et à l'évolution des systèmes hybrides (alternance d'évolutions en temps continus et de remises à niveau en temps discret) qui commencent à être étudiées tant en économie qu'en automatique.

L'objet de cette théorie mathématique est d'offrir une boîte à outils permettant l'étude de l'évolution de systèmes. Elle permet donc de rechercher les conditions (décisions, états) dans lesquelles les contraintes opérationnelles (comme celles induites par le contexte ou l'utilisation des ressources) seront toujours satisfaites et donc dans lesquelles le système pourra fonctionner de manière durable.

Dans la suite, on dira que :

– l'évolution d'un système est discrète si l'état, le contrôle et l'évaluation sont considérés à des intervalles de temps réguliers (discrets), ce qui implique que la dynamique est définie par une relation de récurrence,

– l'évolution d'un système est continue si l'état, le contrôle et l'évaluation sont considérés à chaque instant, la dynamique est alors donnée par une équation ou une inclusion différentielle.

2.1. *Introduction à la théorie de la viabilité*

Comme un système, en évoluant, doit en permanence s'adapter au contexte, il devra pour rester viable, pouvoir réagir de façon opportune à la fois à la présence d'obstacles et/ou à des événements. Il faut donc anticiper le moment où l'état du système atteint ces limites de viabilité. Résoudre cette question, c'est déterminer le « Noyau de Viabilité » du système, soit déterminer le plus grand domaine, dans lequel les variables du système doivent impérativement se maintenir pour permettre au système de fonctionner.

Pour introduire plus finement la théorie de la viabilité, plaçons-nous dans le cas élémentaire de systèmes dynamiques continus de la forme :

$$x'(t) = f(x(t), u(t)) \text{ pour presque tout } t > 0 \qquad [1]$$

dont l'état est contraint par un ensemble K (i.e. $x(t) \in K$), où u représente un paramètre de contrôle choisi dans un ensemble $U(x(t))$. Cet ensemble K peut être défini par un système classique de contraintes statiques (e.g. $K = \{(x, y) \mid a \leq x \leq b, c \leq y \leq d\}$).

Nous allons réécrire le système (1) sous la forme d'une inclusion différentielle :

$$\begin{cases} x' \in F(x) \text{ pour presque tout } t > 0 \\ F(x) := f(x, U(x)) := \{f(x, u(x))\}_{u \in U(x)} \end{cases} \quad [2]$$

En fait, une inclusion différentielle du type (2) est la généralisation de la notion d'équation différentielle aux fonctions multivoques[1], ceci peut ainsi permettre de modéliser l'incertitude que l'on a sur les modèles du problème.

Etant donné un système dynamique défini par (2) et un ensemble K défini par un système de contraintes indépendantes du temps, il n'y a aucune raison pour qu'à partir d'un point initial $x_0 = x(0)$, il existe une solution de (2) qui reste toujours dans K ou au moins jusqu'à un instant T. On s'intéresse alors aux solutions viables de (2), i.e. aux solutions qui ne violent jamais les contraintes imposées, c'est à dire qui restent toujours dans K.

Définition 2.1 *Une fonction $x(.)$ est dite* viable *dans K si $\forall t \in [0, T],\ x(t) \in K$.*

Le plus grand sous-ensemble de valeurs initiales de K à partir duquel est issu au moins une solution viable (ie., qui ne sort jamais de K ou pas avant un instant supérieur à T) est appelé "Noyau de Viabilité" noté $ViabF(K)$. Dans [AUB 91], le théorème suivant a été démontré :

Théorème 2.1 *Soit F une fonction de Marchaud[1] et K un ensemble fermé, alors les conditions suivantes sont équivalentes :*

– K est un domaine de viabilité pour F (ie. $\forall x_0 \in K$ il existe une solution viable de (2) dans K avec la condition initiale $x(0) = x_0$)

– $\forall x \in K, F(x) \cap T_K(x) \neq \emptyset$, c'est à dire $\exists u \in U(x)$ tel que $f(x, u) \in T_K(x)$ où $T_K(x)$ est le cône contingent introduit par Bouligand dans les années 30 pour étendre la notion de "tangente" aux applications multivoques [AUB 90].

Ce théorème admet une version discrète [SAI 98, SAI 94] à partir de laquelle se construit l'algorithme de viabilité.

Dans la section suivante, nous allons tout d'abord considérer le problème de l'algorithme d'approximation du noyau de viabilité d'un système discret dynamique, c'est à dire défini par un système d'inclusions récurrentes sous contraintes. Ensuite nous considérons celui de l'approximation du noyau de viabilité pour un système dynamique continu, c'est à dire défini par un système d'inclusions différentielles sous contraintes.

1. Rappelons qu'une fonction F multivoque, est une application multivoque définie par son graphe $Graph(F) := \{(x, y) \mid y \in F(x)\}$

2.2. *Algorithmes de calcul du noyau de viabilité pour les systèmes dynamiques*

Un système dynamique discret est défini par une relation de récurrence soit une inclusion de type :

$$x_{n+1} \in G(x_n) \qquad [3]$$

Une solution du système (3) est *viable* dans K, s'il existe au moins une suite $(x_0, x_1, \ldots, x_n, \ldots)$, telle que $\forall n,\ x_{n+1} \in G(x_n)$ et $\forall i,\ x_i \in K$. Alors les conditions suivantes sont équivalentes :

– K est un domaine de viabilité discret pour G i.e. tous ses point sont viables

– $\forall x \in K,\ G(x) \cap K \neq \emptyset$

Lorsque K n'est pas un domaine de viabilité discret, la proposition suivante montre l'existence d'un noyau de viabilité contenu dans K et donne une méthode constructive permettant son approximation. Pour cela, on introduit la suite décroissante d'ensembles K_n à partir de $K_0 = K$ et de la relation de récurrence :

$$K_{n+1} := \{x \in K_n \ \text{ tels que } G(x) \cap K_n \neq \emptyset\}\,. \qquad [4]$$

Proposition 2.1 ([AUB 91]) *La limite (au sens de Painlevé-Kuratowski)*[2] *de la suite des ensembles K_n est le noyau de viabilité discret de K pour G. C'est à dire*

$$Viab_G(K) := \bigcap_{j=1}^{\infty} K_j \qquad [5]$$

Il est alors trivial, que dans le cas fini, cet algorithme (défini par $K_0 = K$ et la relation de récurrence (4) s'arrête au pas n sur l'ensemble K_n avec soit K_n vide soit $K_n = K_{n-1}$, ensemble qui mathématiquement correspond au noyau de viabilité de K pour le système dynamique discret (3). L'implémentation naïve de cet algorithme, consiste donc à itérer sur les ensembles K_n jusqu'à ce que la précédente condition d'arrêt "$K_n = \emptyset$ ou $K_n = K_{n-1}$" soit satisfaite.

La deuxième méthode possible est de calculer le noyau de viabilité non pas comme le domaine de viabilité le plus grand mais comme l'ensemble regroupant tous les points à partir desquels est issu une solution viable. Dans le cas continu on peut trouver les propriètes necessaires dans [MAT 96].

On note par $\vartheta_F(t, x_0)$, l'ensemble des valeurs de $x(t)$, au moment t, qui sont accessibles à partir de $x_0 \in K$ par les solutions de (2). Pour un sous-ensemble $X \in K$ on peut définir $\vartheta_F(t, X) = \cup_{x_0 \in X} \vartheta(t, x_0)$.

2. La limite supérieure d'une suite d'ensembles au sens de Painlevé est définie comme $\limsup_{n\to\infty} X_n := \{x \in K \mid \liminf_{n\to\infty} d(x, X_n) = 0\}$ où $d(x, X)$ est la distance entre le point x et le sous-ensemble X defini comme $\inf_{y\in X} d(x, y)$ [AUB 90]

Définition 2.2 *Considèrons une solution à l'inclusion différentielle (2)* $x(.)$*. On appelle le sous-ensemble de ses points d'adherence* $\omega_F(x(.)) := \limsup_{t\to+\infty}\{x(t)\}$ *avec* $t \to +\infty$*,* l'ensemble $\omega-$limite *pour* $x(.)$*.*

Proposition 2.2 (Mattioli et al., [MAT 96]) *On considère une application Marchaud* $F : X \rightsquigarrow X$*. On définit pour tout* $x \in K$ *et pour toute solution* $x(.) \in \vartheta_F(x) \cap K$ *l'ensemble :*

$$\mathbb{E}_F(x(.)) := \left(\cup_{t\geq 0} \{x(t)\} \right) \cup \omega_F(x(.))$$

qui est un domaine de viabilité fermé et

$$Viab_F(K) = \bigcup \left\{ E \in \bigcup_{x\in K} \bigcup_{x(.)\in\vartheta_F(x)} \mathbb{E}_F(x(.)) \;\middle|\; E \subset K \right\} \qquad [6]$$

Cette propriété nous permet d'implémenter un autre algorithme pour calculer le noyau de viabilité pour un système dynamique discret. Les deux méthodes donnent le même résultat et on peut choisir (4) ou (6) suivant les propriétés du problème à résoudre. La différence entre ces méthodes est très similaire aux algorithmes de parcours en largeur ou en profondeur d'abord d'un graphe fini.

Pour utiliser ces algorithmes dans le cas fini, on aura besoin d'une méthode permettant d'approcher le noyau de viabilité continu par une suite des noyaux de viabilité discrets qui sera decrite dans la section suivante.

2.3. *Approximation du noyau de viabilité*

Afin d'obtenir une approximation discrète du noyau de viabilité continu, on choisit un pas de temps ε et on associe à ε un pas d'une trame X_h de X qui désigne le produit cartésien des variables continues du problème. On munit X de la norme $\|x\| = sup\|x_i\|$ où x_i est la i-ème variable, et on note par $\mathcal{B}$ la boule unité associée. Alors les deux assertions suivantes doivent être satisfaites :

– Pour tout compact $K \subset X$, $K_h = K \cap X_h$ possède un nombre fini d'éléments.

– $\forall h > 0,\ \forall x \in X, \exists x_h \in X_h$ tel que $\|x - x_h\| \leq \frac{h}{2}$

A l'application multivoque F, on associe l'application discrétisée approchée $G_{\varepsilon,h}$ définie sur X_h et dépendante des pas de temps et de la taille de la trame. Les details de l'approximation sont introduits dans [SAI 94].

Proposition 2.3 ([SAI 94]) *Dans le cas d'une application* $\ell-$*Lipschitz on peut approcher* F *par une suite des applications* $G_{\varepsilon,h}$ *definie par*

$$G_{\varepsilon,h}(x) \;:=\; \left[x + \varepsilon F(x) + (2h + \ell\varepsilon h + M\ell\varepsilon^2)\mathcal{B}\right] \cap X_h \qquad [7]$$

qui satisfait les conditions necessaires pour la convergence de la suite $Viab_{G_{\varepsilon,h}}K$ *vers* $Viab_F(K)$ *quand* $\varepsilon, h \to 0^+$*.*

L'approximation ainsi définie nous permet d'approcher le noyau de viabilité pour une inclusion continue $x' \in F(x)$. On utilisera cette approximation pour construire une contrainte globale qui permettra de modéliser les systèmes hybrides dans le cadre de la programmation par contraintes.

3. Une contrainte globale pour aborder les systèmes hybrides (statique + dynamique)

Cette section sera consacrée à la description d'une contrainte globale de "viabilité" qui, en combinaison avec les autres concepts de la programmation par contraintes, permettra de modéliser les systèmes complexes par un ensemble de variables aux domaines définis et un ensemble des relations algèbriques (les contraintes). Cette approche permettra de modéliser les systèmes dynamiques utilisant les contraintes statiques.

En effet, la PPC aborde le problème de la description de systèmes complexes par une modélisation formelle des composants réels qui les définissent, constituant par eux-mêmes la réalisation. Un programme en contraintes est donc vu comme un ensemble de modèles, un pour chaque constituant qu'il soit fonctionnel ou physique. Les spécifications étant exprimées à partir de relations, un modèle est donc identifié à l'ensemble des relations définies sur ses variables. Formellement, à tout problème P on se donnera un triplet $((v_1 \dots v_n)\ ;\ (dom(v_1), \dots, dom(v_n))\ ;\ (\phi_1, \dots, \phi_n))$, où les v_i sont les variables à instancier par une valeur du domaine $dom(v_i)$ de telle sorte que les formules ϕ_j (les contraintes) soient satisfaites. Dans toute la suite, on suppose que ces formules sont soit des formules algébriques soit des équations ou inclusions différentielles comme celles définies par (1) dans le cadre continu ou par (3) dans le cadre discret.

Même si un programme en PPC peut utiliser des algèbres différentes pour l'expression de ses contraintes, elles utilisent un processus commun la propagation de contraintes, processus qui peut permettre de détecter rapidement une inconsistance ou de réduire les domaines des variables. Formellement, on peut définir la propagation d'un ensemble de contraintes comme un opérateur algébrique permettant de récupérer dans les domaines, l'information contenue dans les contraintes.

Rappelons qu'un opérateur ϕ sur un treillis complet $\mathcal{L}$ est une application de $\mathcal{L}$ dans lui-même. On dit que $\mathcal{L}$ est

– *croissant* si $\forall x, y \in \mathcal{L}, x \leq y \Rightarrow \phi(x) \leq \phi(y)$

– *extensif (resp. anti-extensif)* ssi $\forall x \in \mathcal{L}, x \geq \phi(x) (resp. x \leq \phi(x)$

– *idempotent* ssi $\forall x \in \mathcal{L},\ \phi(x) = \phi\left(\phi(x)\right)$

Définition 3.1 *Une* ouverture algébrique *(resp. une* fermeture algébrique*) est un opérateur croissant, anti-extensif et idempotent (resp. croissant, anti- extensif et idempotent).*

Dans [APT 98] et [APT 99] , la propagation est définie comme une ouverture algébrique sur le treillis produit cartésien des domaines, opérateur qui maintient certaines formes de cohérence locale. Lorsque l'on se place dans le treillis dual de l'information contenue, on peut montrer que la propagation est une fermeture algébrique. A chaque réduction de domaine, le système a une information plus fine sur la variable.

3.1. *Propriétés algébriques du noyau de viabilité*

Montrons tout d'abord que l'opérateur $Viab_F$ qui transforme un ensemble fermé K en son noyau de viabilité $Viab_F(K)$, possède les bonnes propriétés algébriques sous-jacentes aux mécanismes de propagation. En effet, pour toute fonction F (discrète ou non), $Viab_F$ est une ouverture algébrique dans le treillis des fermés muni de l'union et de l'intersection.

– Dans ([MAT 96]) on montre que $Viab_F$ est un fermé.

– $Viab_F$ est anti-extensivif par définition ($\forall K,\ Viab_F(K) \subset K$)

– Supposons que $K \subset L$ alors $\forall x_0 \in Viab_F(K)$ il existe une solution $x(.)$ viable dans K i.e. $\forall t,\ x(t) \in K \subset L$. Alors, $x_0 \in Viab_F(L)$ et $Viab_F(K) \subset Viab_F(L)$. L'operateur $Viab_F$ est donc croissant.

– Par croissance et anti-extensivité on a $Viab_F(Viab_F(K)) \subset Viab_F(K)$. Soit $x_0 \in Viab_F(K)$. , alors il existe une solution $x(.)$, viable dans K. S'il existe un t_0 tel que $x(t_0) \notin Viab_F(K)$ alors il existe $t_1 \geq 0$ tel que $x(t_0 + t_1) \notin K$. Mais, $x(.)$ est une solution viable et cela est impossible. Alors, $Viab_F$ est idempotent.

– $Viab_F$ est une ouverture algebrique comme un operateur anti-extensif, croissant et idempotant.

Par construction, l'algorithme (4) présenté dans le paragraphe 2.2 est un algorithme itératif qui converge en décroissant soit vers le noyau de viabilité discret (resp. vers un sur-ensemble qui contient le noyau de viabilité continu l'approximation étant d'autant plus précise que h est petit). Ces deux algorithmes sont donc bien des algorithmes de propagation qui suppriment incrémentalement des valeurs initiales de K à partir desquelles il n'existe pas de solution viable du système dynamique.

3.2. *Une contrainte globale de viabilité*

L'approche contrainte globale permet un niveau d'abstraction assez élevé pour modéliser des parties du problème. La puissance d'expression de ces contraintes minimise ainsi la distance entre le besoin de l'utilisateur et le langage de modélisation. Elles englobent des algorithmes de propagation spécifiques.

Dans toute la suite, on se place dans le cadre de la PPC en domaine fini. Considérons donc le problème P défini par le triplet $((v_1 \ldots v_n)$; $(dom(v_1), \ldots, dom(v_n))$;

$(\phi_1, \ldots, \phi_n))$ où les v_i sont des variables à valeurs dans des domaines finis $dom(v_i)$ et ϕj les contraintes du problème. La modélisation de P se subdivise donc en deux parties disctinctes :

– La modélisation des contraintes statiques du système. Par exemple, les contraintes sous-jacentes sont des formules algébriques.

– La modélisation de la dynamique du système caractérisée par un ensemble K et un système G d'inclusions différentielles portant sur une liste de variables, incluse dans l'ensemble des variables de P.

L'ensemble K est alors modélisé comme un ensemble de points défini par un sous-système de contraintes, de même, G est un ensemble de relations de récurrence que l'on peut soit transformer en contraintes du langage PPC sous-jacent soit si cela est trop complexe en simples fonctions. On peut ainsi introduire une contrainte globale de viabilité dont la syntaxe est la suivante :

$$\boxed{\text{Viab (}\ Lvar_K, Lconstr_K, < Dfinition\ de\ G >\text{)}}$$

où $Lconstr_K$ est la liste des contraintes définissant l'ensemble K et $Lvar_K$ est la liste des variables associées, et $< Dfinition\ de\ G >$ correspond soit à l'ensemble des contraintes soit à la fonction modélisant le système dynamique discret G.

La contrainte Viabilité a pour sémantique :

Il existe au moins un point viable pour le système G dans le domaine courant des variables satisfaisant le système défini par $Lconstr_K$

Ce qui exprime le système d'équation suivant :

$$\forall(X, \ldots) \in Lvar_k,\ \exists(x_0, \ldots) \in ((Dom(X) \times \ldots) \mid (x_0, \ldots) \in Viab_G(K)$$

Par exemple, dans le cas d'un ensemble K à deux dimensions, cette contrainte sera vraie si et seulement si $\exists(x_0, y_0) \in Dom(X) \times Dom(Y)$ tel que $(x_0, y_0) \in Viab_G(K)$.

Une autre utilisation possible de cette contrainte utilise le résultat suivant [CAR 94] : l'épigraphe de la fonction de temps minimal d'atteinte d'une cible d'un problème de contrôle optimal coïncide avec un noyau de viabilité. Dans ce cas on peut donc aisément dériver de la contrainte Viab une contrainte de temps minimal T d'atteinte d'une cible C comme :

$$\boxed{\text{OptTime (}\ Lvar_K, T, Lconstr_K, Lconstr_C, < Definition\ of\ G >\text{)}}$$

En d'autre terme T est le temps minimal pour atteindre la cible C avec une dynamique contrôlée par le système G tout en restant viable dans K. On montrera dans la section 4 un exemple d'utilisation de cette variante de la contrainte de viabilite.

Une première implémentation de ces deux contraintes globales a été réalisée dans le solveur PPC Eclair©[MUS 03], bibliothèque de contraintes sur les domaines finis, développée par Thales depuis 1997 et construite au-dessus de Claire©.

3.3. *Cohérence maintenue par les contraintes*

Dans la section précédente nous avons décrit deux nouvelles contraintes permettant de prendre en compte la dynamique dans un contexte statique d'un système de contraintes. Ces deux contraintes peuvent propager l'information inhérente à la dynamique du système initial. Ils peuvent non seulement signaler l'absence d'une solution dans les domaines de ses variables mais aussi maintenir une sorte de cohérence sur ces domaines. Comme dans le cas des contraintes classiques, il existe plusieurs types de la cohérence à maintenir. On montrera la propagation assurée sur l'exemple de la contrainte de viabilité sachant que la contrainte de temps optimal est un cas particulier de la contrainte de viabilité.

Dans le cas géneral on peut définir la cohérence forte qui assure le faite que chaque valeur dans chaque domaine peut participer à l'affectation qui satisfait cette contrainte :

$$\begin{array}{c} \forall X \in Lvar,\ \forall x_0 \in Dom(X),\ \exists(\ldots, y_0, \ldots) \in (\ldots \times Dom(Y) \times \ldots) \\ \text{tel que } (\ldots, x_0, \ldots, y_0, \ldots) \in Viab_G(K) \end{array} \quad [8]$$

On peut aussi relacher cette condition et définir une cohérence plus faible. Cette variante de cohérence est souvent appellée la cohérence aux bornes. Le principe sous-jacent est d'omettre la vérification de viabilité des valeurs entre les bornes d'un domaine si les valeurs sur ces bornes sont viables. Cela peut être le cas pour la variable correspondante à la dimension temporelle du problème s'il s'agit de la contrainte de temps optimal. Dans ce cas, on vérifie seulement que les bornes du domaine sont viables :

$$\begin{array}{c} \forall X \in Lvar,\ \forall x_0 \in \left\{\min_{x \in Dom(X)} x, \max_{x \in Dom(X))} x\right\}, \\ \exists(\ldots, y_0, \ldots) \in (\ldots \times Dom(Y) \times \ldots) \\ \text{tel que } (\ldots, x_0, \ldots, y_0, \ldots) \in Viab_G(K) \end{array} \quad [9]$$

Dans les deux cas la propagation élimine, à partir des domaines, les valeurs qui ne sont pas cohérents avec la contrainte. Cela réduit l'espace de recherche qui sera exploré pendant la résolution de la partie combinatoire du problème.

Ces contraintes peuvent maintenir la cohérence d'arc dans le cas des contraintes $p-$aires ou seulement la cohérence aux bornes de ces domaines. En utilisant de telles contraintes, on pourrait modéliser les problèmes pour lesquels les variables de décision sont aussi utilisées pour définir les conditions initiales d'un problème dynamique.

On montrera la propagation effectuée sur un exemple numérique d'une des applications possibles dans la section suivante.

4. Exemple numérique

Pour illustrer l'utilisation des contraintes de viabilité nous avons choisi le problème de nageur décrit par Zermelo. La déscription formelle de cet exemple et les exemples

d'application de la théorie de la viabilité pour les problèmes de contrôle se trouvent dans [CAR 94] et [CAR 98].

Considérons le problème suivant, dérivé du problème de navigation de Zermelo. Une petite barque de pêcheurs navigue sur un plan d'eau (lac, rivière, ...) à une vitesse indépendante de la direction dont la norme maximale est supposée dépendante du temps $c(t)$. Le courant est donné par un champ de vecteurs, fonction de la position. La vitesse du courant décroit près des rives et donnée par l'equation

$$f(x,y) := \left\{1 - a|y^2|\right\} \times \{0\} \qquad [10]$$

Le pêcheur désire gagner une île avec sa barque. Pour y arriver sans dommage, il doit absolument éviter des obstacles (banc de sable, zone de courant.). Il s'agit de déterminer les positions (x_0, y_0) à partir de desquelles il existe une trajectoire viable atteignant l'île en temps fini. L'ensemble K représente donc la zone navigable du plan d'eau. Il est défini comme $[-6, 2] \times [-5, 5] \times [0, 30]$ et on a ajoute de contraintes supplémentaires representant les obstacles. Ces obstacles sont définis par les inégalités lineaires (dont un, par exemple, occupe la partie gauche de la figure 1). La cible est définie comme $C = \left\{(x, y) \in \mathbb{R}^2 | x^2 + y^2 \leq 0.44\right\}$. La dynamique globale du nageur peut être decrite comme

$$(x'(t), y'(t)) = f(x, y) + cu \qquad [11]$$

où u prend les valeurs dans la boule unité de $\mathbb{R}^2$.

Le problème consiste à calculer le temps minimal necessaire pour aller jusqu'à la cible à partir de tout point de K. Si ce temps depasse une certaine borne alors le points initial correspondant n'est pas viable. Tous les autres points sont viables et il existe une solution à l'inclusion (11) qui atteint la cible.

En utilisant la contrainte de temps minimal (ou la contrainte de viabilité) on peut modéliser les problèmes définis comportant des sous-problèmes dynamiques. Pour donner un exemple d'un tel problème on peut choisir un problème combinatoire avec plusieurs nageurs avec les dynamiques et cibles différentes.

La figure 1 montre les courbes de niveau de l'approximation de la fonction temps minimal. Pour illustrer la propagation nous avons ajouté un rectangle transparent qui represent les domaines des variables X et Y de la contrainte de viabilité (et correspondant à l'ensemble $Dom(X) \times Dom(Y)$). La partie opaque de ce rectangle montre les valeurs qui sont eliminés à partir du domaine de la variable X (par (8) ou (9)). Ces valeurs ne peuvent pas participer dans aucune solution du problème. Même dans le cas continus ces points ne seront pas viables a cause de sur-approximation qui est faite pendant la discrétisation. Le même type de propagation peut être appliqué à la variable T désignant le temps necessaire pour atteindre la cible (dans le cas de la contrainte du temps optimal).

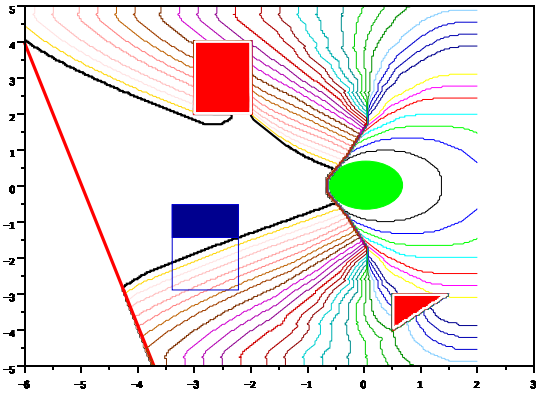

Figure 1 – Les courbes de niveau pour le problème de Zermelo avec les obstacles

5. Conclusions et perspectives

Les problèmes d'aide à la décision sous-jacents à des systèmes complexes comme les systèmes de supervision (SICs civil ou militaire), la gestion de portefeuilles (finance), etc., sont souvent de nature combinatoire et possèdent aussi des contraintes dynamiques. La modélisation de tels systèmes met donc en jeu des variables et des contraintes de natures diverses : des variables à domaine discret à valeurs numériques ou non, des variables à valeurs réelles, des contraintes linéaires, numériques et dynamiques (e.g. régies par des équations différentielles). De plus, en fonction de la quantité et de la qualité des informations disponibles (et de leur degré de confiance), ces systèmes doivent raisonner sur des données imprécises ou incomplètes. De manière générale, un problème de décision consiste en un choix (ou un classement) parmi une liste d'alternatives possibles, compte tenu de connaissances que le décideur (ou le système) a sur l'état du monde et de ses préférences ou de l'objectif à atteindre. Les systèmes d'aide à la décision abordés par Thales présentent à des degrés divers une ou plusieurs caractéristiques fondamentales parmi les suivantes :

– une nature hybride (continu vs. discret, statique vs. dynamique),

– un niveau d'indétermination (lié aux incertitudes et aux imprécisions),

– un niveau de complexité élevé (induisant des sous-problèmes combinatoires avec contraintes, des problèmes de couplages et de régulation de ces sous-systèmes),

– l'exigence d'une certaine qualité des solutions.

L'analyse de ces systèmes complexes est aujourd'hui un domaine de recherche très actif dans les milieux académiques et industriels. Issue à l'origine d'une problé-

matique à cheval entre l'automatique et l'informatique, la recherche dans ce domaine a été menée, jusqu'à présent, principalement au sein de ces deux disciplines. En effet, la plupart du temps, ces systèmes sont modélisés par des automates munis de variables typées par des entiers, des réels, des chaînes de caractères, des arbres, etc. Cependant ils dépendent de façon cruciale du développement et de la maintenance des logiciels qui les supportent et pour cette raison, il est nécessaire de mettre en œuvre une démarche générique de modélisation qui pourra être enrichie par des bibliothèques de modèles spécifiques. Ce type d'approche accroît la réutilisation en permettant le développement d'applications multiples à partir d'un effort de modélisation dédié à un domaine unique.

Par ailleurs, ces systèmes hybrides dynamiques ne bénéficient actuellement pas de la technologie "contrainte" car elle est très peu adaptée à la résolution de ce type de problème. Cependant, nous estimons que la PPC fournit un cadre particulièrement adéquat pour aborder la *modélisation* de tels systèmes. En effet, le formalisme "contraintes" est suffisamment unificateur pour permettre la composition de nouveaux modèles issus de divers concepts émergents (arithmétiques d'intervalles, théorie de la viabilité, les algèbres exotiques comme (max,+) .).

Au travers de cet article, nous avons donc confirmé le pouvoir de modélisation de la PPC grâce à l'introduction de la notion de "noyau de viabilité" pour prendre en compte les lois d'évolution du système. En effet, un problème hybride dynamique/statique est modélisé au travers d'un système d'équations (voire d'inclusions) différentielles et d'un système de contraintes ou directement [AUB 91] comme un problème dynamique de recherche de noyau de viabilité sous contraintes. En couplant une version incrémentale de l'algorithme de viabilité (discret ou approximation du continu) avec des mécanismes d'arc cohérence du solveur, il est donc possible de concevoir une contrainte globale qui permet de propager ainsi les informations inhérentes à la dynamique du système afin de réduire l'espace de recherche. Une première implémentation a été réalisée dans l'environnement Eclair©.

Enfin, plusieurs sortes d'évolutions peuvent être adressées :

– *une évolution déterministe* modélisée par une équation différentielle ;

– *une évolution dans le risque* : L'équation différentielle régissant l'évolution de la situation dépend d'un ou de plusieurs paramètres qualifiant la "probabilité" ou la "possibilité" de réalisation d'un événement ;

– *une évolution dans l'incertain* : L'équation différentielle régissant l'évolution de la situation dépend d'un ou de plusieurs paramètres de perturbation inconnus et est souvent modélisée par le biais d'une inclusion différentielle.

6. Bibliographie

[APT 98] APT K. R., « The Essence of Constraint Propagation », *CWI Quarterly*, vol. 11, n° 2 & 3, 1998, p. 215-248.

[APT 99] APT K. R., « The Rough Guide To Constraint Propagation », JAFFAR J., Ed., *CP'99*, vol. 1713 de *LNCS*, Springer-Verlag, 1999, p. 1-24.

[AUB 84] AUBIN J.-P., CELLINA A., *Differential inclusions (Set-valued maps and Viability Theory*, Springer-Verlag, 1984.

[AUB 90] AUBIN J.-P., FRANKOWSKA H., *Set-Valued Analysis*, Birkhäuser, 1990.

[AUB 91] AUBIN J.-P., *Viability Theory*, Birkhäuser, 1991.

[AUB 97] AUBIN J.-P., *Dynamic Economic Theory : a Viability Approach*, Springer-Verlag, 1997.

[AUB 98] AUBIN J.-P., NAJMAN L., « The Russian Mountain Algorithm for Global Optimisation », *J. Mathematical methods of Operations Research*, , 1998.

[AUB 99] AUBIN J.-P., *Mutational and Morphological Analysis : Tools for Shape Regulation and Morphogenesis*, Birkhäuser, 1999.

[BON 97] BONNEUIL N., MULLERS K., « Viable Populations in a predator-prey system », *J. Mathematical Biology*, vol. 35, 1997, p. 261-293.

[CAR 94] CARDALIAGUET P., QUINCAMPOIX M., SAINT-PIERRE P., « Optimal times for constrained controlled systems without local controllability », *C. R. Acad. Sci. Paris*, vol. 318, n° 1, 1994, p. 607-612.

[CAR 98] CARDALIAGUET P., QUINCAMPOIX M., SAINT-PIERRE P., « Set-Valued Numerical Analysis for Optimal Control and Differential Games », M.BARDI, T.PARTHASARATHY, RAGHAVEN T., Eds., *Stochastic and Differential Games*, Birkhäuser, 1998.

[DOY 96] DOYEN L., GABAY D., « Economie des ressources renouvellables et viabilité », *Actes des journées Vie, Environnement et Sociétés*, 1996.

[DOY 97] DOYEN L., GABAY D., « Viabilité et Régulation d'un modèle de croissance prenant en compte le risque climatique », *Journées du PIRPVS, Toulouse*, 1997.

[GOR 96] GORRE A., « The "Montagnes Russes" Algorithm in mutational spaces », *Parametric Optimization IV*, , 1996, Springer-Verlag.

[MAT 96] MATTIOLI J., DOYEN L., NAJMAN L., « Lattice operators underlying dynamic systems », *Set-Valued Analysis*, vol. 4, 1996, p. 119-134.

[MUS 03] MUSEUX N., JEANNIN L., SAVÉANT P., HUÉDÉ F. L., FRANCOIS-XAVIER JOSSET, MATTIOLI J., « Claire/Eclair : Un environnement de modélisation et de résolution pour des applications d'optimisations combinatoires embarquées », *JFPLC-03*, 2003.

[QUI 91] QUINCAMPOIX M., « Problèmes de cibles en théorie du contrôle et des jeux différentiels », PhD thesis, Paris Dauphine, 1991.

[SAI 94] SAINT-PIERRE P., « Approximation of Viability Kernel », *Applied Mathematics and Optimisation*, vol. 29, 1994, p. 187-209.

[SAI 98] SAINT-PIERRE P., « Dynamical System with State Constraints, Theory and Applications », *IFIP Conference on Systems Modelling and Optimization*, Addison Wesley Longman, 1998.