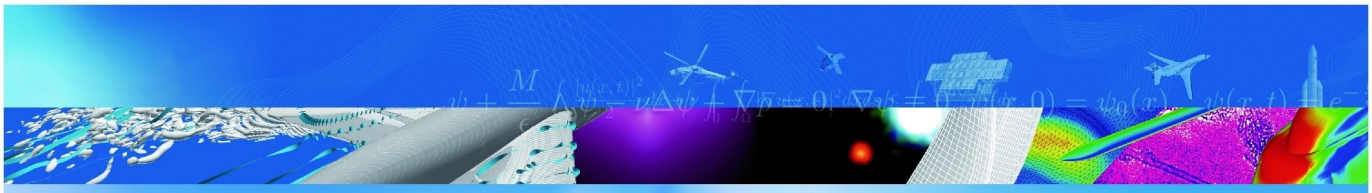
Synthèse de plans d'exécution multi-agents robustes aux incertitudes et à l'absence de communications

G. Casanova, C. Pralet, C. Lesire Cabaniols, Thierry Vidal



COMMUNICATION A CONGRES

Synthèse de plans d'exécution multi-agents robustes aux incertitudes et à l'absence de communications

G. Casanova (ONERA), C. Pralet (ONERA),
C. Lesire (ONERA), T. Vidal (ENIT)

JFSMA 2016

ROUEN, FRANCE

5-10 octobre 2016

TP 2016-655

70 2016 ans

ONERA

THE FRENCH AEROSPACE LAB

Synthèse de plans d'exécution multi-agents robustes aux incertitudes et à l'absence de communications

G. Casanova[1] Guillaume.Casanova@onera.fr
C. Pralet[1] Cedric.Pralet@onera.fr
C. Lesire[1] Charles.Lesire@onera.fr
T. Vidal[2] Thierry.Vidal@enit.fr

[1]Onera – The French Aerospace Lab
F-31055, Toulouse, France

[2]Ecole Nationale d'Ingenieurs de Tarbes,
Tarbes, France

Résumé

L'exécution de missions multi-agents nécessite de gérer les incertitudes induites par des événements incertains. Quand les communications sont intermittentes, il est nécessaire pour les agents d'agir selon leur vue locale du problème, de manière indépendante aux événements contrôlés ou observés par les autres agents. Dans ce papier, nous proposons un nouveau cadre de travail pour aborder cette problématique, en particulier pour les plans de missions impliquant des contraintes temporelles. Ce cadre de travail, appelé Réseau Temporel Simple Multi-agents avec Incertitudes (MaSTNU), est une combinaison entre les cadres Réseau Temporel Simple Multi-agents (MaSTN) et Réseau Temporel Simple avec Incertitudes (STNU). Nous définissons la propriété de contrôlabilité dynamique pour les MaSTNU ainsi qu'une méthode pour calculer hors-ligne des stratégies d'exécution valides qui sont ensuite distribuées aux agents. Cette méthode est basée sur une formulation en Programmation Linéaire en Nombres Entiers et peut également être utilisée pour optimiser certains critères telle que la flexibilité temporelle de plans d'exécution multi-agents.

Mots-clés : *Contrôlabilité Dynamique, Planification Multi-agents, MaSTNU, PLNE*

Abstract

Executing multi-agent missions requires managing the uncertainty about uncontrollable events. When communications are intermittent, it additionally requires for each agent to act only based on its local view of the problem, that is independently of events which are controlled or observed by the other agents. In this paper, we propose a new framework for dealing with such contexts, with a focus on mission plans involving temporal constraints. This framework, called Multi-agent Simple Temporal Network with Uncertainty (MaSTNU), is a combination between Multi-agent Simple Temporal Network (MaSTN) and Simple Temporal Network with Uncertainty (STNU). We define the dynamic controllability property for MaSTNU, and a method for computing offline valid execution strategies which are then dispatched between agents. This method is based on a mixed-integer linear programming formulation and can also be used to optimize criteria such as the temporal flexibility of multi-agent plans.

Keywords: *Dynamic Controllability, Multi-agent Planning, MaSTNU, MIP*

1 Introduction

Dans les applications robotiques telles que l'exploration autonome de larges zones dangereuses, de meilleures performances peuvent être atteintes par l'utilisation de plusieurs robots. En effet, cela peut mener à des temps d'exécution de la mission plus courts grâce à la parallélisation des tâches à effectuer, et à une augmentation de la résilience aux pannes du système grâce à la redondance des compétences des robots. Une difficulté à surmonter dans ce contexte est que les tâches allouées aux robots doivent être coordonnées, à cause de l'existence de contraintes de précédence ou de synchronisation entre les différentes tâches, ou plus généralement de contraintes sur les distances temporelles minimales ou maximales entre les tâches. Afin de gérer ces contraintes temporelles multi-agents, les *Réseaux Temporels Simples Multi-agents* (Multi-agent Simple Temporal Networks ou MaSTN [1]) ont récemment été introduits, ainsi que des techniques pour calculer de manière distribuée les distances temporelles entre les événements du plan [1], ou les dates d'exécution au plus tôt/tard des tâches [2].

Cependant, l'utilisation des MaSTN pour des missions en robotique présente un obstacle majeur : les MaSTN ne sont pas prévus pour obtenir des stratégies de décision robustes aux in-

certitudes sur les dates d'occurrence des événements incontrôlables. Par exemple, les MaSTN ne sont pas adaptés pour obtenir des plans d'exécution réalisables quelles que soient les durées exactes des tâches à l'exécution. Ainsi, ils ne sont pas aussi expressif que le cadre de travail *Réseau Temporel Simple avec Incertitudes* (Simple Temporal Networks with Uncertainties ou STNU [8]), qui fait une distinction claire entre les points temporels dits *exécutables*, qui sont directement contrôlés, et les points temporels dits *contingents*, qui ne le sont pas. Avec la formulation STNU, les stratégies d'exécution robustes décrivent une manière de choisir les dates d'exécution des points temporels exécutables en fonction des dates d'exécution des points temporels (exécutables ou contingents) observés précédemment. Ces stratégies sont construites en supposant que l'exécution de chaque point temporel dans le réseau temporel est instantanément observée. Une telle hypothèse est souvent contredite dans les systèmes multi-robots, puisque certains événements ne peuvent être observables qu'à partir d'un ensemble restreint de positions géographiques, voir être observables uniquement par un ensemble restreint d'agents.

C'est pourquoi nous proposons un nouveau cadre de travail pour gérer les contraintes temporelles pour les systèmes multi-agents. Ce cadre de travail, appelé *Réseau Temporel Simple Multi-agents avec Incertitudes* (Multi-agent Simple Temporal Networks with Uncertainties ou MaSTNU), peut être vu comme une tentative de combiner les MaSTN et les STNU. Nous définissons pour ce cadre MaSTNU des algorithmes pour calculer des stratégies d'exécution robustes respectant chaque contrainte temporelle d'un plan multi-agents, en prenant en compte les incertitudes sur les dates d'exécution des points temporels contingents. Les stratégies produites sont également applicables en environnements contraints induisant des communications intermittentes ou inexistantes. De telles stratégies d'exécution distribuées sont obtenues en utilisant hors-ligne une procédure centralisée basée sur une formulation en programmation linéaire en nombres entiers (PLNE) du problème que nous appelons *problème de contrôlabilité dynamique multi-agents*. Cette procédure est exécutée au niveau du centre de mission avant le déploiement coordonné des agents sur le terrain.

Ce papier est organisé comme suit. La section 2 introduit des rappels sur les cadres STN, MaSTN et STNU. La section 3 présente le cadre de travail MaSTNU. La section 4 détaille notre approche PLNE pour résoudre le problème de contrôlabilité multi-agents. La section 5 discute certaines métriques d'optimisation sur les stratégies d'exécution.

2 Contexte

2.1 Réseau Temporel Simple (STN)

Les *Problèmes Temporels Simples* (Simple Temporal Problems ou STP [4]) sont un cadre de travail couramment utilisé pour raisonner sur les contraintes temporelles. Un STP est une paire $S = (V, E)$ définie par un ensemble $V = \{v_1, \ldots, v_n\}$ de points temporels représentant les dates d'occurrence des événements, et un ensemble E de contraintes temporelles reliant ces points temporels. Chaque contrainte $e \in E$ est de la forme $v_j - v_i \in [L_{ij}, U_{ij}]$, où $L_{ij} \in \mathbb{R} \cup \{-\infty\}$ et $U_{ij} \in \mathbb{R} \cup \{+\infty\}$ spécifient respectivement une distance temporelle minimale et maximale entre v_i et v_j. Un point temporel spécifique v_0 appelé le *point de référence* est habituellement ajouté à V afin de représenter une position temporelle de référence. Les contraintes temporelles unaires telles que $v_i \in [L_{0i}, U_{0i}]$ peuvent ainsi être exprimées sous la forme de contraintes de distance par rapport à ce point de référence (contraintes $v_i - v_0 \in [L_{0i}, U_{0i}]$).

Les STP ont une représentation graphique naturelle appelée *Réseau Temporel Simple* (Simple Temporal Networks ou STN), contenant un sommet pour chaque point temporel dans V et un arc $v_i \rightarrow v_j$ étiqueté par $[L_{ij}, U_{ij}]$ pour chaque contrainte temporelle dans E. Les STN sont utiles en pratique car de nombreux problèmes exprimables sous la forme d'un STN sont solubles en temps polynomial [4, 10, 9], par exemple pour déterminer s'il existe une affectation de tous les points temporels satisfaisant toutes les contraintes temporelles.

2.2 Réseau Temporel Simple Multi-agents (MaSTN)

Les STN ont été étendus au contexte multi-agents, dans lequel les points temporels ne sont pas contrôlés par un agent seul mais sont répartis parmi un ensemble d'agents $\mathcal{A}$. Cette extension est appelée *Réseau Temporel Simple Multi-agents* (Multi-agent Simple Temporal Network ou MaSTN [1]). De manière formelle, un MaSTN est défini par :

— un ensemble de STN *locaux* (un par agent $a \in \mathcal{A}$) ; le STN *local* associé à l'agent a, noté S_L^a est défini par V_L^a, l'ensemble des points temporels *locaux* possédés par a, et E_L^a, l'ensemble d'arcs *locaux* reliant uniquement des points temporels dans V_L^a ; les variables dans V_L^a sont dites *possédées* par a ;
— un ensemble d'arcs *externes* E_X, chacun contraignant les distances temporelles entre deux points temporels appartenant à des agents différents ; en plus de ses arcs locaux dans E_L^a, chaque agent a est supposé connaître les contraintes externes reliées à l'un de ses sommets locaux.

Des algorithmes sur les MaSTN existent pour calculer, de manière distribuée, les distances entre les paires de points temporels [1] ainsi que les dates d'occurrence au plus tôt/tard des points temporels [2].

2.3 Réseau Temporel Simple avec Incertitudes (STNU)

Les STN ne sont pas adaptés pour représenter les durées dites *incertaines*, c'est-à-dire les durées dont les valeurs sont fixées par des processus externes plutôt que par l'agent planificateur lui-même. C'est pourquoi les STN ont été étendus aux *Réseaux Temporels Simples avec Incertitudes* (Simple Temporal Networks with Uncertainties ou STNU [8]).

De manière formelle, un STNU est un triplet (V, E, C), où V est un ensemble de points temporels, E est un ensemble de liens *d'exigence*, et C est un ensemble de liens *contingents*. Les liens d'exigence sont identiques aux arcs du modèle STN standard. Chaque lien contingent est défini par une paire de points temporels (v_i, v_j) et par un intervalle temporel $[L_{ij}, U_{ij}]$ avec $0 < L_{ij} < U_{ij} < \infty$. Dans ce cas, v_i et v_j sont respectivement appelés le point temporel *d'activation* et le point temporel *contingent*. L_{ij} et U_{ij} sont respectivement les bornes inférieure et supérieure de la durée du lien. En outre, un point temporel ne peut être le point temporel contingent de deux liens contingents distincts. Chaque point temporel qui n'est le point temporel contingent d'aucun lien contingent est dit *exécutable*. Dans la suite, nous notons V_E l'ensemble des points temporels exécutables et V_C l'ensemble des points temporels contingents.

Le problème fondamental associé aux STNU est de déterminer s'ils sont *dynamiquement contrôlables* (DC), ce qui signifie de manière informelle qu'il existe une manière d'affecter dynamiquement les valeurs des points temporels exécutables en fonction des observations effectuées, de telle sorte que tous les liens d'exigence soient satisfaits quelle que soit la durée des liens contingents à l'exécution.

Plus formellement, la contrôlabilité dynamique des STNU peut être définie comme suit. Tout d'abord, une *projection* d'un STNU est un STN obtenu en remplaçant chaque lien contingent $v_j - v_i \in [L_{ij}, U_{ij}]$ par un lien déterministe $v_j - v_i \in [d, d]$ avec $d \in [L_{ij}, U_{ij}]$. Une *programmation* est une affectation de valeurs à tous les points temporels. Une *stratégie d'exécution* R peut alors être définie comme une fonction associant les projections aux programmations. Une stratégie d'exécution R est dite valide si pour toute projection p, la programmation $R(p)$ satisfait tous les liens d'exigence. Une stratégie d'exécution R est dite dynamique si et seulement si pour tout point temporel exécutable v et pour toutes projections p_1, p_2 du STNU, si les affectations de tous les points temporels programmés avant v sont les mêmes dans $R(p_1)$ and $R(p_2)$, alors les valeurs assignées à v dans $R(p_1)$ et $R(p_2)$ sont les mêmes. Pour finir, le STNU est *dynamiquement contrôlable* si et seulement si il existe une stratégie d'exécution à la fois valide et dynamique.

La figure 1 présente un exemple d'un STNU dynamiquement contrôlable, pour lequel nous pouvons définir une stratégie d'exécution valide : (a) exécuter v_1^A au temps 0, (b) attendre que v_1^B survienne, (c) exécuter v_2^B au temps $v_1^B + 4$, (d) exécuter v_2^A au temps $v_2^B + 6$, (e) exécuter v_3^B au temps $v_2^B + 6$, (f) attendre que v_3^A survienne.

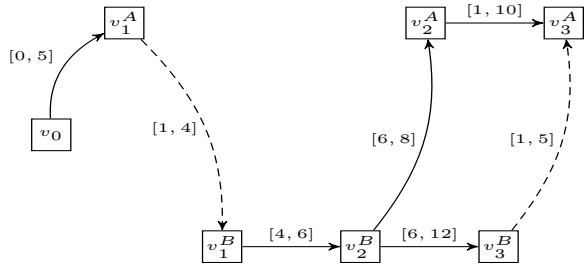


FIGURE 1 – Exemple de STNU dynamiquement contrôlable, les arcs continus représentent les liens d'exigences, les arcs discontinus représentent les liens contingents

2.4 Vérification de la Contrôlabilité Dynamique par utilisation de la PLNE

De nombreux algorithmes existent pour vérifier la contrôlabilité dynamique (DC) d'un STNU [8, 6, 5]. L'un d'eux consiste à calculer des contraintes *d'attentes* obligatoires sur les liens d'exigence. Une contrainte d'attente (v_k, w_{ijk}) sur un lien d'exigence $(v_i, v_j) \in E$ signifie que v_j peut uniquement être exécuté ou bien après que v_k a été exécuté, ou bien après w_{ijk} unités de temps depuis l'exécution de v_i.

Dans une approche différente, la vérification de la DC des STNU peut également être formulée sous la forme d'un Programme Linéaire en Nombres Entiers (PLNE) [3]. L'un des avantages d'une telle formulation est qu'elle peut être adaptée pour répondre à des requêtes plus générales, telle que la minimisation des changements de bornes sur les liens contingents de telle sorte qu'un STNU non DC le devienne.

La figure 2 présente le modèle linéaire disjonctif introduit dans [3], à partir duquel un modèle PLNE peut être obtenu après quelques étapes de linéarisation. Le modèle contient deux variables de décisions continues l_{ij} et u_{ij} pour chaque paire de points temporels (v_i, v_j). Les variables l_{ij} et u_{ij} représentent respectivement les bornes inférieures et supérieures imposées sur la distance $v_j - v_i$ entre les points temporels v_i et v_j. Ces bornes sont introduites même si aucune contrainte explicite n'existe entre v_i et v_j Le modèle contient également un ensemble de variables d'attente continues w_{ijk} (une variable par triplet de points temporels (v_i, v_j, v_k) où v_k est un point temporel contingent). Ces variables d'attente ont la même signification que les contraintes d'attente vues précédemment. Des variables de décision discrètes sont présentes dans le modèle PLNE après le processus de linéarisation. Si une solution au problème est trouvée, alors le STNU est DC. Voir [3] pour les détails concernant la validité de la modélisation et le processus de linéarisation.

Une fonction d'optimisation f_{opt} peut aisément être ajoutée au modèle, par exemple pour maximiser la flexibilité des solutions en utilisant $f_{opt} = \sum_{i<j}(u_{ij} - l_{ij})$.

$$\forall (v_i, v_j) \in E, L_{ij} \leq l_{ij} \leq u_{ij} \leq U_{ij} \tag{1}$$

$$\forall (v_i, v_j) \in C, (l_{ij} = L_{ij}) \wedge (u_{ij} = U_{ij}) \tag{2}$$

$$\forall v_i, v_j, v_k \in V^3, \begin{cases} l_{ik} \leq u_{ij} + l_{jk} \leq u_{ik} \\ l_{ik} \leq l_{ij} + u_{jk} \leq u_{ik} \\ u_{ik} \leq u_{ij} + u_{jk} \\ l_{ij} + l_{jk} \leq l_{ik} \end{cases} \tag{3}$$

$$\forall (v_i, v_k) \in C, \forall v_j \in V_E, (l_{jk} < 0) \vee \begin{pmatrix} u_{ij} \leq l_{ik} - l_{jk} \\ l_{ij} \geq u_{ik} - u_{jk} \end{pmatrix} \tag{4}$$

$$\forall (v_i, v_k) \in C, \forall v_j \in V_E, u_{ik} - u_{jk} \leq w_{ijk} \tag{5}$$

$$\forall (v_i, v_j) \in E, \forall v_k \in V_C, \min(l_{ik}, w_{ijk}) \leq l_{ij} \tag{6}$$

$$\forall (v_i, v_k), (v_m, v_j) \in C^2, (w_{ijk} < 0) \vee (w_{ijk} - l_{mj} \leq w_{imk}) \tag{7}$$

$$\forall (v_i, v_k) \in C, \forall v_m, v_j \in V^2, w_{ijk} - u_{mj} \leq w_{imk} \tag{8}$$

FIGURE 2 – Modèle linéaire disjonctif pour encoder DC sur les STNU

3 Réseau Temporel Simple Multi-agents avec Incertitudes (MaSTNU)

3.1 Définition du cadre de travail

Comme expliqué en introduction, le cadre STNU ne peut pas être directement réutilisé dans le contexte multi-agents, dans lequel chaque agent contrôle uniquement un sous-ensemble des points temporels et n'observe que les exécutions d'un sous-ensemble des points temporels. C'est pourquoi nous introduisons le cadre des STNU Multi-agents (MaSTNU).

De manière formelle, un MaSTNU est un quadruplet $(\mathcal{A}, V, E, C)$, avec $\mathcal{A}$ un ensemble d'agents et (V, E, C) un STNU. V représente l'ensemble des points temporels exécutables et contingents, E l'ensemble des liens d'exigence, et C l'ensemble des liens contingents. De plus, de même que pour les MaSTN (cf. section 2.2), les points temporels dans V sont partitionnés dans $\mathcal{A}$, c'est-à-dire que pour chaque point temporel $v \in V$ il existe un agent unique $a \in \mathcal{A}$ qui *possède* v, noté $owner(v) = a$. Si v est un point temporel exécutable, alors le propriétaire de v est l'agent contrôlant l'exécution de l'événement associé à v. Le point de référence v_0 représente une horloge partagée entre les agents, considérée comme simultanément possédée par tous les agents.

Dans la suite, pour chaque agent $a \in \mathcal{A}$, V^a re-

présente l'ensemble des points temporels possédés par a, qui sont appelés les points temporels *locaux* de a. Nous notons E_L^a (respectivement C_L^a) l'ensemble des liens d'exigence (respectivement liens contingents) *locaux*, connectant des points temporels possédés par a. De manière analogue aux MaSTN, nous définissons également E_X (respectivement C_X) comme l'ensemble des liens d'exigence (respectivement liens contingents) *externes*, connectant deux points temporels possédés par des agents distincts.

La figure 3 présente un exemple de MaSTNU impliquant deux agents A et B, possédant respectivement les points temporels $V^A = \{v_1^A, v_2^A, v_3^A\}$ et $V^B = \{v_1^B, v_2^B, v_3^B\}$. Le lien partant de v_1^A à v_1^B est un lien contingent externe, le lien partant de v_2^B à v_2^A est un lien d'exigence externe, le lien partant de v_1^B à v_2^B est un lien d'exigence local, et il n'y a pas de lien contingent local.

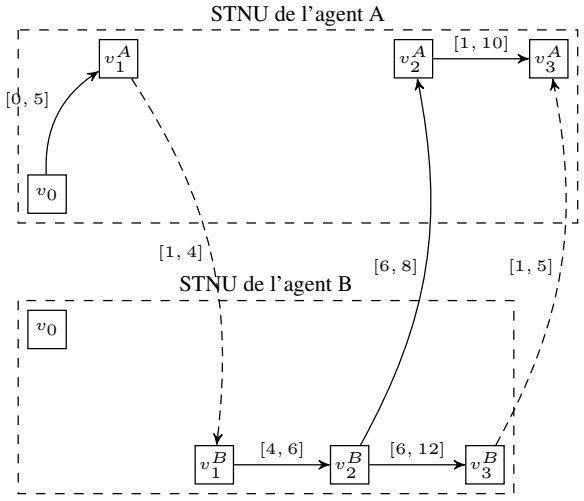


FIGURE 3 – Exemple de MaSTNU

3.2 Contrôlabilité Dynamique Multi-agents

La nature multi-agents des MaSTNU nécessite une adaptation de la propriété de contrôlabilité dynamique. En effet, pour un MaSTNU $(\mathcal{A}, V, E, C)$, le calcul d'une stratégie d'exécution valide pour le STNU (V, E, C) ne permet pas forcément d'obtenir une stratégie d'exécution multi-agents applicable. Par exemple, considérons le MaSTNU présenté à la figure 3. Le STNU associé est le STNU précédemment vu à la figure 1. En considérant la stratégie d'exécution déjà présentée pour ce STNU et en la distribuant entre les différents agents, nous obtenons la stratégie suivante :

- pour l'agent A : (a) exécuter v_1^A au temps 0, (b) exécuter v_2^A au temps $v_2^B + 6$, (c) attendre que v_3^A survienne ;
- pour l'agent B : (a) attendre que v_1^B survienne, (b) exécuter v_2^B au temps $v_1^B + 4$, (c) exécuter v_3^B au temps $v_2^B + 6$.

Le problème d'une telle stratégie est que l'agent A ne possède aucune garantie sur la validité de son exécution, parce qu'il se pourrait qu'il n'observe pas, ou avec un délai trop important, le point temporel externe v_2^B possédé par l'agent B.

C'est pourquoi nous introduisons une nouvelle définition de la contrôlabilité dynamique, adaptée aux MaSTNU. Soit $(\mathcal{A}, V, E, C)$ un MaSTNU et R une stratégie d'exécution pour le STNU associé (V, E, C). La stratégie d'exécution R est dite *distribuée* si et seulement si pour toutes projections p, p' et pour tout agent $a \in \mathcal{A}$, si les programmations $R(p)$ et $R(p')$ affectent toutes deux les mêmes valeurs à tous les points contingents possédés par a, alors elles affectent également les mêmes valeurs à tous les points temporels exécutables possédés par a. En d'autres termes, chaque agent agit uniquement en fonction de ses propres observations immédiates, ce qui signifie que la stratégie d'exécution est robuste face aux éventuelles observations manquantes des points temporels externes.

Un MaSTNU $(\mathcal{A}, V, E, C)$ est alors dit *dynamiquement contrôlable* (DC) si est seulement si le STNU (V, E, C) admet une stratégie d'exécution qui soit valide, dynamique et distribuée.

4 Vérification de la contrôlabilité dynamique et calcul de stratégies d'exécution

Pour vérifier si un MaSTNU $(\mathcal{A}, V, E, C)$ est DC, nous vérifions d'abord si le STNU (V, E, C) est DC. Si ce STNU n'est pas DC, alors le MaSTNU ne l'est pas non plus, car les stratégies d'exécution valables pour le MaSTNU sont plus restrictives que les stratégies d'exécution valables pour le STNU. Dans le cas où le STNU (V, E, C) est DC, nous effectuons des opérations supplémentaires pour déterminer si le MaSTNU original est DC.

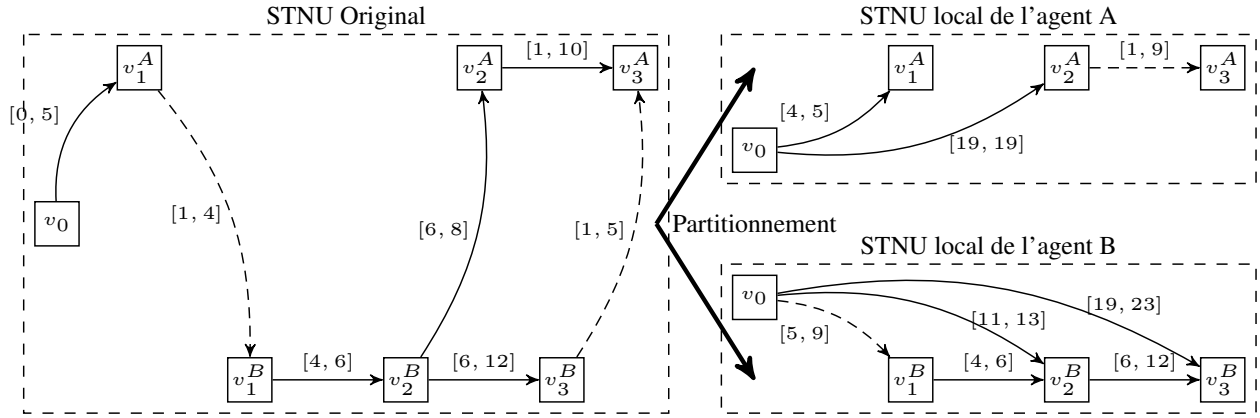


FIGURE 4 – MaSTNU original et son partitionnement

4.1 D'un MaSTNU à un ensemble de STNU locaux

L'idée principale de notre méthode est de transformer le MaSTNU original S en un MaSTNU *distribué*, ne contenant aucun lien externe inter-agents, puis de partitionner ce MaSTNU distribué en un ensemble de STNU locaux $\{S^a \mid a \in \mathcal{A}\}$. La figure 4 illustre ce processus. La raison pour laquelle nous cherchons à obtenir un MaSTNU distribué est que si chaque agent $a \in \mathcal{A}$ utilise une stratégie d'exécution valide R^a pour son propre STNU local S^a, alors la stratégie globale obtenue en combinant les stratégies R^a est valide et dynamique. De plus, cette stratégie globale est également distribuée car nous sommes sûrs grâce à R^a que chaque agent agit uniquement sur la base des observations qu'il est censé obtenir à l'exécution, sans possibilité d'être influencé par des points temporels extérieurs grâce au partitionnement. En d'autres termes, l'ensemble des stratégies d'exécution locales $\{R^a \mid a \in \mathcal{A}\}$ permet de contrôler dynamiquement le MaSTNU.

Pour transformer le MaSTNU original en un ensemble de STNU locaux, nous devons effectuer deux types d'opérations :

1. remplacer les liens d'exigence externes du MaSTNU original par des liens d'exigence locaux aux agents, comme fait dans la figure 4 pour (v_2^B, v_2^A) ; les liens d'exigence locaux introduits peuvent être plus restrictifs que dans le MaSTNU original, pour des besoins de coordinations entre les actions des agents ;
2. supprimer les liens contingents externes et les remplacer par des liens contingents locaux, comme fait dans la figure 4 pour (v_1^A, v_1^B) ; c'est-à-dire que la source externe v d'un lien contingent (v, w) doit être remplacée par une source locale u appartenant à l'agent possédant w.

Le processus réalisant ces deux opérations est présenté dans les deux sections suivantes. Contrairement à la vérification de la contrôlabilité dynamique pour les STNU, il est important de noter que la transformation du MaSTNU original en plusieurs STNU locaux est un problème de décision combinatoire, par exemple à cause du fait qu'il n'y a pas une seule manière de distribuer ou de partager la satisfaction des liens d'exigence externes parmi les agents, ou encore une seule manière de réassigner la source d'un point temporel contingent. Le problème de décision associé est formalisé en utilisant un modèle PLNE, ce qui nous permet de réutiliser des éléments du modèle PLNE déjà existant donné à la section 2.4. Dans le modèle PLNE ainsi défini, nous capturons des contraintes garantissant la satisfaction des liens d'exigence externes du MaSTNU original, ainsi que plusieurs contraintes garantissant que les hypothèses de contingence effectuées pour le MaSTNU distribué ne sont pas restrictives par rapport à l'ensemble des scénarios couverts par les liens contingents externes du MaSTNU original. En ajoutant une fonction d'optimisation linéaire, les solveurs PLNE peuvent alors être utilisés pour obtenir une distribution optimale des contraintes temporelles.

Dans le reste de cet article, de même que dans le modèle PLNE pour vérifier DC sur les STNU, nous utilisons pour tous $i < j$ les variables l_{ij} et u_{ij} pour représenter les bornes au plus tôt/tard imposées sur la distance $v_j - v_i$ entre v_i et v_j.

De plus, pour $i < j$, nous utilisons également u_{ij} comme substitut pour $-l_{ij}$.

4.2 Internalisation des liens d'exigence externes

Considérons un lien d'exigence externe $e = (v_i, v_j)$, avec $owner(v_i) = a, owner(v_j) = b, a \neq b$. Initialement, e est étiqueté par $[L_{ij}, U_{ij}]$. Cependant, a et b peuvent ne pas avoir suffisamment d'informations durant l'exécution pour assurer que e soit respecté. Par exemple, si b attend d'observer v_i avant d'exécuter v_j alors il est possible que e ne soit pas respecté à l'exécution si le délai pour observer v_i est plus grand que U_{ij}. Inversement, a ne possède aucune information lui indiquant quand il doit exécuter v_i de sorte que b ait la possibilité d'exécuter v_j tout en respectant e.

Afin d'assurer que la borne au plus tard de e soit respectée durant l'exécution même sans communication entre a et b, il suffit de trouver un chemin $p = \{p_1, ..., p_k\}$, composé de points temporels, tel que :

1. le chemin p définit un chemin de v_i à v_j plus court que U_{ij}, ainsi si toutes les contraintes entre les points temporels dans p sont satisfaites, alors e est également satisfait ;
2. chaque lien (p_i, p_{i+1}) impliqué dans p est soit un lien interne ($owner(p_i) = owner(p_{i+1})$), soit un lien contingent externe ($(p_i, p_{i+1}) \in C_X$ ou $(p_{i+1}, p_i) \in C_X$).

De tels chemins sont appelés *chemins distribués*. Un lien contingent présent dans un chemin distribué est nécessairement satisfait par définition. Un lien d'exigence interne dans un chemin distribué est satisfiable à l'exécution par l'agent le possédant à condition que le STNU local soit DC. Par conséquent, si tous les STNU locaux sont DC, alors toutes les contraintes impliquées dans le chemin p sont satisfiables à l'exécution. Des chemins distribués similaires doivent également être trouvés afin d'assurer que la borne au plus tôt de e soit satisfaite.

L'un des obstacles dans la modélisation de la recherche d'un chemin distribué garantissant la satisfaction d'un lien d'exigence externe est que le nombre de chemins possibles est exponentiel par rapport au nombre de points temporels. Afin de conserver un modèle compact et de laisser le solveur chercher parmi les différents chemins envisageables, il est possible de représenter les chemins distribués sous la forme d'un ensemble de triangles représentant la triangulation du chemin.

Dans les équations ci-dessous, nous définissons $\overline{E_X}$ l'ensemble de liens d'exigence externes implicites : $\overline{E_X} = \{(v_i, v_j) | owner(v_i) \neq owner(v_j)\} \wedge (v_i, v_j) \notin C_X\}$. Nous définissons également T l'ensemble de *triangles externes* : $T = \{(v_i, v_j, v_k) | (v_i, v_j) \in \overline{E_X}, v_k \in V \setminus \{v_i, v_j\}\}$.

Afin de modéliser la condition que tous les liens d'exigence externes soient couverts par des chemins distribués, nous utilisons dans notre modèle PLNE les variables suivantes :

- $\forall (v_i, v_j) \in \overline{E_X}, b_{ij} \in \{0, 1\}$ sont des variables de décisions booléennes encodant le besoin de justifier un lien d'exigence externe $v_j - v_i \leq u_{ij}$;
- $\forall (v_i, v_j, v_k) \in T, t_{ijk} \in \{0, 1\}$ sont des variables de décisions booléennes encodant que nous utilisons le triangle (v_i, v_j, v_k) (c'est-à-dire le chemin $v_i \rightarrow v_k \rightarrow v_j$) pour justifier la satisfaction du lien d'exigence externe $v_j - v_i \leq u_{ij}$;
- $\forall (v_i, v_j) \in \overline{E_X}, h_{ij} \in [0, |V| - 1]$ sont des variables de décisions encodant la *hauteur* de la justification de la satisfaction de $v_j - v_i \leq u_{ij}$; ces variables de hauteur sont utilisées pour éviter les cycles, par exemple pour éviter les cas où la satisfaction d'une borne supérieure d'un lien externe e est justifiée par la borne supérieure associée au lien externe e', et pour lequel la borne supérieure de e' est justifiée par la borne supérieure de e.

Nous imposons plusieurs contraintes linéaires afin de représenter la satisfaction des liens d'exigence externes par des chemins distribués. Tout d'abord, pour chaque lien contingent, les bornes inférieure et supérieure associées à ce lien sont les mêmes que dans le MaSTNU original :

$$\forall (v_i, v_j) \in C_X, \ (l_{ij} = L_{ij}) \wedge (u_{ij} = U_{ij}) \quad (9)$$

Chaque lien d'exigence externe initial doit être justifié :

$$\forall (v_i, v_j) \in E_X \text{ t.q. } U_{ij} \neq +\infty, \ b_{ij} = 1 \quad (10)$$
$$\forall (v_i, v_j) \in E_X \text{ t.q. } L_{ij} \neq -\infty, \ b_{ji} = 1 \quad (11)$$

Si un lien d'exigence externe doit être justifié, alors un triangle le justifiant doit exister :

$$\forall (v_i, v_j) \in \overline{E_X},\ b_{ij} \leq \sum_{v_k \in V \setminus \{v_i, v_j\}} t_{ijk} \quad (12)$$

Un lien d'exigence externe $v_j - v_i$ est justifié par le triangle (v_i, v_j, v_k) si le chemin $v_i \rightarrow v_k \rightarrow v_j$ est plus court que le chemin $v_i \rightarrow v_j$:

$$\forall (v_i, v_j, v_k) \in T, u_{ij} \geq u_{ik} + u_{kj} + (t_{ijk} - 1)M \quad (13)$$

Dans l'équation précédente, M est une constante égale à $(L_{ij} - U_{ik} - U_{kj})$, de telle sorte que la contrainte soit toujours satisfaite quand $t_{ijk} = 0$.

Chaque lien d'exigence externe dans un triangle doit être justifié :

$$\forall (v_i, v_j, v_k) \in T \text{ t.q. } (v_i, v_k) \in \overline{E_X},\ t_{ijk} \leq b_{ik} \quad (14)$$

$$\forall (v_i, v_j, v_k) \in T \text{ t.q. } (v_k, v_j) \in \overline{E_X},\ t_{ijk} \leq b_{kj} \quad (15)$$

Pour finir, nous empêchons les cycles dans les justifications :

$$\forall (v_i, v_j, v_k) \in T,$$
$$\begin{cases} (v_i, v_k) \in \overline{E_X} : h_{ij} + (1 - t_{ijk})|V| \geq h_{ik} + 1 \\ (v_k, v_j) \in \overline{E_X} : h_{ij} + (1 - t_{ijk})|V| \geq h_{kj} + 1 \end{cases} \quad (16)$$

La figure 5 montre en action le processus d'internalisation de liens d'exigence externes. Dans cet exemple, A et B contractent les contraintes internes (v_0, v_1^A), (v_0, v_2^A), (v_0, v_2^B) et (v_0, v_3^B) afin que le lien d'exigence externe (v_2^B, v_2^A) soit satisfait à l'exécution. Contrairement à la solution du STNU mono-agent vue en figure 1, v_2^A ne possède plus de flexibilité temporelle : la distance temporelle par rapport à v_0 est fixée à la valeur 19.

4.3 Internalisation des liens contingents externes

Chaque lien contingent externe doit être internalisé, dans le cas contraire des points temporels externes pourraient apparaître dans les stratégies d'exécution locales, invalidant la distributivité de ces stratégies. L'idée principale de l'internalisation des liens dans C_X est que

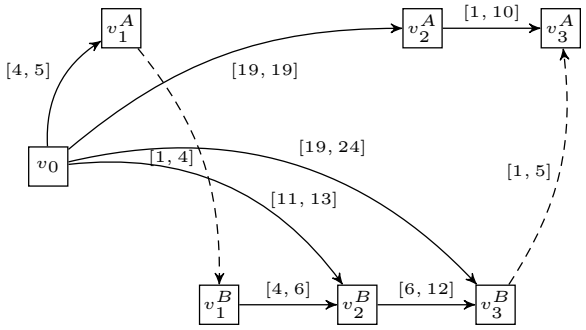


FIGURE 5 – Internalisation de liens d'exigences externes

chaque situation potentielle pouvant être rencontrée dans le MaSTNU originel doit être couverte par les scénarios considérés au niveau des STNU locaux.

Afin d'illustrer la transformation proposée, considérons un lien contingent externe $c = (v_i, v_j)$, avec $owner(v_i) = a, owner(v_j) = b, a \neq b$. c est étiqueté par $[L_{ij}, U_{ij}]$, avec $L_{ij} > 0$. Toute stratégie d'exécution utilisant le fait que "v_j survient nécessairement entre L_{ij} et U_{ij} unités de temps après v_i" ne peut être valable puisque b n'observe pas directement v_i. C'est pourquoi nous devons explicitement éliminer c de la représentation MaSTNU tout en maintenant v_j en tant que point temporel incontrôlable. La seule solution est de remplacer le lien (v_i, v_j) par un lien contingent interne (v_k, v_j) dans l'ensemble des contraintes contingentes locales de l'agent b. Dans ce cas, nous disons que nous utilisons le triangle de substitution (v_i, v_j, v_k). Dans la suite, nous définissons l'ensemble des triangles de substitution candidats par $Q = \{(v_i, v_j, v_k) \,|\, (v_i, v_j) \in C_X, v_k \in V^{owner(v_j)} \setminus \{v_j\}\}$. Pour chaque lien contingent externe (v_i, v_j), à cause des multiples choix possibles de source v_k pour activer v_j, nous ajoutons au modèle PLNE les variables de décisions suivantes :

— $\forall (v_i, v_j, v_k) \in Q$, $c_{kj} \in \{0, 1\}$ est une variable de décision booléenne encodant que nous substituons le lien contingent externe (v_i, v_j) par un nouveau lien contingent interne (v_k, v_j).

Plusieurs contraintes sont imposées sur ces variables. Tout d'abord, chaque lien contingent externe doit être substitué par exactement un lien contingent interne :

$$\forall (v_i, v_j) \in C_X, \sum_{v_k \,|\, (v_i, v_j, v_k) \in Q} c_{kj} = 1 \quad (17)$$

Si un lien contingent externe (v_i, v_j) est substitué par un lien contingent interne (v_k, v_j), alors les bornes spécifiées par (v_i, v_j) ne doivent pas être moins restrictives que les bornes données par le chemin $v_i \rightarrow v_k \rightarrow v_j$:

$$\forall (v_i, v_j, v_k) \in Q, \begin{cases} u_{kj} \geq u_{ki} + u_{ij} + (c_{kj} - 1)M \\ 0 < l_{kj} \leq l_{ki} + l_{ij} + (1 - c_{kj})M \end{cases} \quad (18)$$

Si le lien contingent externe (v_i, v_j) est substitué par le lien contingent interne (v_k, v_j), alors le lien d'exigence associé (v_i, v_k) doit être justifié :

$$\forall (v_i, v_j, v_k) \in Q, \begin{cases} (v_i, v_k) \in \overline{E_X} : c_{kj} \leq b_{ik} \\ (v_k, v_i) \in \overline{E_X} : c_{kj} \leq b_{ki} \end{cases} \quad (19)$$

La figure 6 montre le MaSTN distribué obtenu après internalisation des liens d'exigence externes ainsi que des liens contingents externes. Comparé à l'exemple précédent, (v_0, v_3^B) a dû être davantage contraint. De plus, (v_0, v_1^B) et (v_2^A, v_3^A) sont désormais des liens contingents.

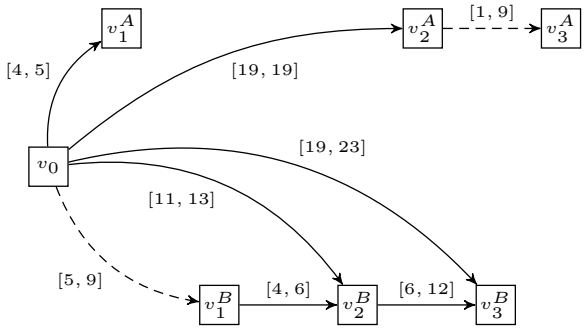


FIGURE 6 – Internalisation des liens externes

4.4 Contrôlabilité dynamique des STNU locaux

Pour finir, nous devons exprimer le fait que les STNU locaux associés à chaque agent doivent être DC. Cette propriété est exprimée en adaptant le modèle vu à la section 2.4. Une adaptation est nécessaire à cause des choix effectués dans le processus d'internalisation des liens externes, en particulier car l'ensemble des liens contingents internes n'est pas fixé. Cela implique par exemple que la contrainte donnée à l'équation 8 doit être remplacée par :

$$\forall v_k \in V_C, \forall v_i \in V^{owner(v_k)} \setminus \{v_k\}, \forall v_m, v_j \in V^2, \\ w_{ijk} - u_{mj} \leq w_{imk} + (1 - c_{ik})M' \quad (20)$$

avec M' une constante suffisamment grande. Des transformations similaires doivent être appliquées pour les équations 4, 5, et 7.

4.5 Distribution des STNU locaux

Si une solution au problème PLNE existe, cette solution décrit un STNU distribué. Ce dernier peut ensuite être partitionné en un ensemble de N STNU locaux, un pour chaque agent, en ignorant les contraintes externes. Les STNU locaux sont alors envoyés aux agents, et la mission peut démarrer. Il faut néanmoins noter que cette technique utilisée pour vérifier DC est valide mais non complète, essentiellement à cause du processus d'internalisation des liens contingents durant lequel des informations de corrélations entre différents points temporels sont perdues.

5 Optimisations supplémentaires

Notre méthode peut optimiser l'ensemble des STNU locaux obtenus en maximisant la fonction objectif $f_{opt} = \sum_{i<j}(u_{ij} - l_{ij})$. De nombreuses autres métriques peuvent être utilisées. Par exemple, la figure 7 présente les STNU locaux obtenus en minimisant le temps d'exécution au plus tard du dernier point temporel à être exécuté, afin de finir la mission le plus rapidement possible (minimisation de $f_{opt} = \max_i u_{0i}$). Avec cette nouvelle fonction objectif, le temps d'exécution de la mission au plus tard est réduit de 19 unités de temps à 16 unités de temps.

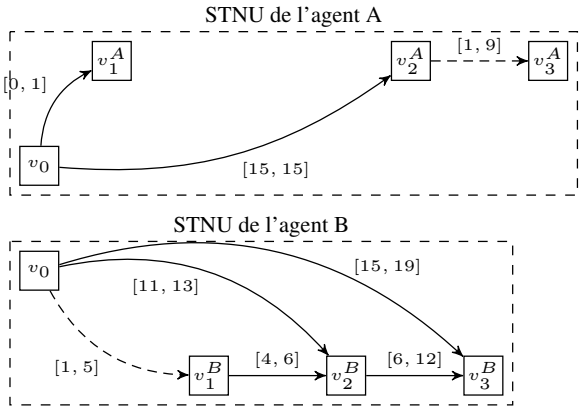


FIGURE 7 – Optimisation du temps d'exécution

Nous avons testé la résolution de notre modèle PLNE en utilisant le solveur CPLEX. La validité de l'approche a été vérifiée sur un ensemble de MaSTNU aléatoirement générés. Toutes les

expérimentations ont été exécutées sur des processeurs Intel à 3.0GHz et 4GB de mémoire vive. La résolution du modèle PLNE prend typiquement entre 0.2 secondes pour l'exemple à 6 nœuds utilisé dans ce papier à 1200 secondes pour un example dense à 4 agents et 40 nœuds possédant 4 liens contingents externes et 10 liens d'exigence externes, ce qui est cohérent avec l'implémentation utilisée qui est basée sur l'algorithme DC-checking en $O(|V|^5)$ trouvé dans [8].

Une première solution dans le dernier cas, améliorant la solution ignorant les liens externes contingents qui nous permettent d'obtenir des solutions plus flexibles, était cependant trouvée en 80 secondes. Ces résultats pourraient être grandement améliorés par l'utilisation d'heuristiques de recherche ou en se limitant à trouver de bonnes solutions au lieu de solutions optimales, ce qui permettrait d'améliorer la faisabilité opérationnelle de cette méthode.

Des algorithmes DC-checking plus efficaces, de complexité $O(|V|^4)$, peuvent être trouvés dans la littérature, cependant ces algorithmes sont plus complexes à traduire en formalisme PLNE.

6 Conclusion

La contrôlabilité dynamique est une propriété importante pour les plans temporels avec incertitudes puisqu'elle permet d'augmenter les chances de succès d'une mission. Dans ce papier, nous avons montré comment gérer les contraintes temporelles incertaines dans les plans temporels multi-agents grâce aux Réseaux Temporels Simples Multi-agents avec Incertitudes, et comment obtenir un plan exécutable partitionné entre les agents par la résolution d'un modèle PLNE. Plusieurs pistes de travail sont envisageables afin d'améliorer la gestion des MaSTNU, par exemple distribuer le processus PLNE de résolution dans le but de réoptimiser les plans temporels durant la phase en ligne, ou prendre en compte des communications lors de rendez-vous spécifiques pendant l'exécution de la mission [7].

Références

[1] James C. Boerkoel, Léon Planken, Ronald Wilcox, and Julie A. Shah. Distributed Algorithms for Incrementally Maintaining Multiagent Simple Temporal Networks. In *Int. Conf. on Automated Planning and Scheduling (ICAPS)*, Rome, Italy, 2013.

[2] Guillaume Casanova, Cédric Pralet, and Charles Lesire. Managing dynamic multiagent simple temporal network. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1171–1179, Istanbul, Turkey, 2015.

[3] Jing Cui, Peng Yu, Cheng Fang, Patrik Haslum, and Brian C. Williams. Optimising bounds in simple temporal networks with uncertainty under dynamic controllability constraints. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS)*, pages 52–60, Jerusalem, Israel, 2015.

[4] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal Constraint Networks. *Artificial Intelligence*, 49(1-3) :61–95, 1991.

[5] Luke Hunsberger. Efficient execution of dynamically controllable simple temporal networks with uncertainty. *Acta Informatica*, 52(8) :1–59, 2015.

[6] Paul Morris. Dynamic controllability and dispatchability relationships. In *Integration of AI and OR Techniques in Constraint Programming - 11th International Conference (CPAIOR)*, pages 464–479, Cork, Ireland, 2014.

[7] Paul H. Morris and Nicola Muscettola. Managing temporal uncertainty through waypoint controllability. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1253–1258, Stockholm, Sweden, 1999.

[8] Paul H. Morris, Nicola Muscettola, and Thierry Vidal. Dynamic Control Of Plans With Temporal Uncertainty. In *Int. Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, WA, USA, 2001.

[9] Léon Planken, Mathijs de Weerdt, and Neil Yorke-Smith. Incrementally Solving STNs by Enforcing Partial Path Consistency. In *Int. Conf. on Automated Planning and Scheduling (ICAPS)*, Toronto, Canada, 2010.

[10] Lin Xu and Berthe Y. Choueiry. A New Efficient Algorithm for Solving the Simple Temporal Problem. In *Int. Symposium on Temporal Representation and Reasoning (TIME)*, Cairns, Queensland, Australia, 2003.

Office National d'Études et de Recherches Aérospatiales
2 avenue Edouard Belin - BP 74025
31055 TOULOUSE Cedex 4
Tél. : +33 5 62 25 25 25
http://www.onera.fr