Ordonnancement et planification sous contraintes temporelles et probabilistes

Bassam Baki

Ordonnancement et planification sous contraintes temporelles et probabilistes

Bassam BAKI
GREYC- Université de Caen
Campus II- Bd. Maréchal Juin
14032 Caen Cedex
bbaki@info.unicaen.fr

Résumé : Nous nous int´eressons à la planification et l'ordonnancement temporels sous incertitude qui permettent de r´esoudre un ensemble des buts.
Nous proposons une approche de planification temporelle permettant aux tâches de respecter des contraintes temporelles et de pr´ec´edence. Dans ce problème, chaque tâche possède un ensemble de contraintes temporelles, un ensemble de probabilit´es d'ex´ecution et un ensemble de coûts. Une relation de pr´ec´edence relie les tâches dans un graphe ET/OU. Grâce à une propagation temporelle à travers le graphe, nous calculons les intervalles d'ex´ecution des tâches. Puis, nous choisissons le meilleur plan pouvant être ex´ecut´e en respectant toutes les contraintes. Le problème que nous traitons dans cet article combine deux types de planification : temporelle et probabiliste.
Mots-clés : Planification, Ordonnancement, Raisonnement temporel, Incertitude, Aide à la d´ecision et Intelligence Artificielle.

1 INTRODUCTION

La planification consiste en la recherche d'une s´equence d'op´erations permettant de passer d'un ´etat initial à un ´etat final souhait´e. Un problème d'ordonnancement consiste à organiser dans le temps un ensemble d'activit´es, de façon à satisfaire un ensemble de contraintes et optimiser une ou plusieurs fonctions objectifs. En d'autres termes, la planification consiste à d´eterminer les tâches à ex´ecuter et l'ordonnancement consiste à d´eterminer quand ex´ecuter ces tâches.
Le contrôle d'ex´ecution de plans est un problème particulièrement difficile lorsqu'il doit être effectu´e à bord de systèmes autonomes tels que des robots. Un tel système doit disposer de processus de d´elib´eration pour engendrer des plans qui r´ealisent les buts de la mission tout en respectant des d´elais et des contraintes de pr´ec´edence. Plusieurs formalismes et algorithmes ont ´et´e d´evelopp´es pour traiter des problèmes de planification tenant compte de contraintes temporelles. De tels problèmes de planification existent dans de nombreuses applications r´eelles telles que le transport, la robotique, le domaine m´edical, les services d'urgences, etc.
Le premier système qui a introduit la repr´esentation des actions par pr´e-conditions/effets est STRIPS [FN71]. Depuis, plusieurs formalismes et algorithmes ont ´et´e d´evelopp´es pour traiter des problèmes de planification tenant compte de contraintes temporelles.
Le domaine de la planification temporelle est sp´ecifi´e par l'environnement de la planification (un ensemble d'´etats, un ensemble d'actions, un ´etat initial, un ensemble de buts et un ensemble de contraintes de pr´ec´edence) et les caract´eristiques temporelles (dur´ees, dates de d´ebut, dates de fin, temps limites,...). La plupart des formalismes [DMP91] permettent la repr´esentation de contraintes temporelles portant, en g´en´eral, sur la dur´ee d'ex´ecution des tâches. Cependant d'autres types de contraintes peuvent être envisag´es. De plus, les applications r´eelles pr´esentent une certaine incertitude sur le temps. Par exemple, dans un problème de transport, la dur´ee d'un trajet n'est pas connue avec exactitude. Il est donc important de pouvoir raisonner sur le temps et sous incertitude.

Diff´erentes approches de planification sous incertitude temporelle ont ´et´e d´evelopp´ees [MMV01], [DKS02] mais elles ne considèrent pas l'incertitude sur les dur´ees d'ex´ecution. Les ´etudes r´ecentes en planification concernent de plus en plus des problèmes r´eels et traitent plusieurs sortes de contraintes [BDM$^+$02], [TVP03], [PG02]. Les plans g´en´er´es par ces systèmes peuvent être consid´er´es comme une extension des plans classiques.
IxTeT et parcPlan [GL94], [LI04] sont des planificateurs qui traitent des donn´ees temporelles dans la gestion de temps qualitative (ordre et relation utilis´es dans la planification) et dans les contraintes quantitatives (temps et dur´ees utilis´es dans l'ordonnancement). La repr´esentation du temps dans IxTeT est celle de l'algèbre d'instants. Ces instants sont des variables symboliques sur lesquelles sont bas´ees les contraintes temporelles. Comme IxTeT, en ce qui concerne la distance entre les instants temporels, notre planificateur travaille sur des contraintes symboliques (pr´ec´edence) et num´eriques (intervalles $[I^-, I^+]$). Mais de plus, il prend en consid´eration les coûts et les probabilit´es d'ex´ecution des plans dans un graphe ET/OU. N´eanmoins il ne traite pas le cas de partage des ressources.
Dans [DHW94] et [KHW94], les auteurs pr´esentent un planificateur appel´e "C-Buridan" qui construit un plan avec des actions probabilistes (productrices) d'information et une ex´ecution contingente. Leur algorithme est

sûr (s'il arrive à son terme, la solution g´en´er´ee est suffisante pour atteindre le but) et complet (il trouve la solution si elle existe), mais ce planificateur ne prend pas en consid´eration les contraintes temporelles locales et globales, ni les contraintes de pr´ec´edence, ni le d´elai entre les tâches. Ses solutions sont satisfaisantes plutôt qu'optimales.

Dans cet article, nous consid´erons un système de planification et d'ordonnancement qui gère des contraintes temporelles complexes sous incertitude. Notre problème est repr´esent´e par un graphe ET/OU où nous voulons trouver un plan des tâches qui satisfait toutes les contraintes (de temps, de pr´ec´edence et de coût) et qui a une grande probabilit´e d'être ex´ecut´e avec temps et coûts r´eduits. A noter que le problème de recherche d'un plan optimal a ´et´e d´emontr´e NP-complet. Nous nous int´eressons à trouver des solutions satisfaisantes plus qu'optimales. Dans cet article, nous consid´erons qu'un agent doit ex´ecuter un ensemble de tâches reli´ees ensemble dans un graphe acyclique orient´e tel que les nœuds correspondent aux tâches et les arêtes sont des contraintes de pr´ec´edenceÉtant donn´e un ensemble de tâches, un ensemble de donn´ees pour chaque tâche (une date de d´ebut, une date de fin, un ensemble de dur´ees, un ensemble de probabilit´es et un ensemble de coûts) une relation de pr´ec´edence et de d´elais entre les tâches, notre approche permet à l'agent de d´eterminer l'ensemble des tâches à ex´ecuter de telle façon qu'en ex´ecutant ces tâches, on atteint le but en respectant toutes les contraintes temporelles et de pr´ec´edence. Autrement dit, ´etant donn´es : la description du domaine (le graphe, les donn´ees temporelles, les coûts, les probabilit´es,...), un ensemble d'´etats initiaux, un ensemble de buts à r´esoudre, et un ensemble de contraintes correspondant à un ordre partiel que l'on souhaite retrouver dans l'ordre de r´ealisation des buts, notre approche permet de d´eterminer l'ensemble des plans solutions puis par affinements successifs, elle trouve les plans admissibles et parmi ces plans elle choisit, selon un op´erateur de pr´ef´erence, le meilleur plan à ex´ecuter.

Notre approche utilise les techniques de propagation de contraintes temporelles [BW00] qui a pour but de simplifier la r´esolution d'un problème. Pratiquement, cela peut consister à retirer du domaine des variables de d´ecision, les valeurs qui n'appartiennent à aucune solution. Ce filtrage ´evite de nombreuses tentatives de r´esolution vou´ees à l'´echec.

Une particularit´e importante du problème consid´er´e est que les tâches sont repr´esent´ees dans un graphe ET/OU et que les dur´ees des tâches sont pond´er´ees par des probabilit´es d'ex´ecution. Cela exprime notamment des incertitudes sur la connaissance exacte de la dur´ee d'ex´ecution des tâches qui ne sera r´eellement connue qu'à l'ex´ecution effective. Dans cet article, nous supposons qu'une tâche s'ex´ecute pendant l'une de ses dur´ees avec la probabilit´e donn´ee pour cette dur´ee, et si pour une raison quelconque la tâche ne r´eussit pas à s'ex´ecuter en l'une de ses dur´ees, on se trouve dans un cas d'´echec total où il faut arrêter l'ex´ecution du plan choisi et demander une replanification.

Ainsi dans le domaine de l'Intelligence Artificielle, notre planificateur est le premier qui traite des contraintes complexes parmi lesquelles les contraintes temporelles et les probabilit´es d'ex´ecution.
Dans la section suivante, nous d´ecrivons quelques d´efinitions g´en´erales dans la section 2. La section 3 d´ecrit le problème à r´esoudre. Dans la section 4, nous d´ecrirons la propagation des intervalles d'ex´ecution puis la propagation des probabilit´es dans la section 5. L'algorithme utilis´e pour trouver le meilleur plan est pr´esent´e dans la section 6. Nous analyserons notre approche dans la section 7. Nous pr´esenterons des tests exp´erimentaux dans la section 8 et nous conclurons dans la section 9.

2 PRÉLIMINAIRES

Dans cette section, nous d´ecrivons la notion d'un agent et d'un graphe de contraintes temporelles.

2.1 Agent de planification

Le but de l'agent de planification est de choisir, g´erer et ordonner un sous-ensemble de tâches à ex´ecuter parmi un ensemble de tâches repr´esent´ees dans un graphe acyclique.

Définition 1 *On appelle données associées à une tâche t la liste $< I_t^-, I_t^+, \Delta_t, Pr_t, C_t >$ où :*
- *I_t^- est la date la plus précoce à laquelle la tâche t peut commencer son exécution et I_t^+ est la date la plus tardive à laquelle la tâche t doit être terminée de sorte que $[I_t^-, I_t^+]$ est la fenêtre temporelle d'exécution de t ;*
- *$\Delta_t = \{d_t^1, d_t^2, ..., d_t^m\}$ est l'ensemble des durées d'exécution possibles de la tâche t ;*
- *$Pr_t = \{pr_t^1, pr_t^2, ..., pr_t^m\}$, où pr_t^j est la probabilité que la tâche t soit exécutée en la durée d_t^j, on a par conséquent $0 \leq pr_t^j \leq 1$ et $\sum pr_t^j = 1$;*
- *$C_t = \{c_t^1, c_t^2, ..., c_t^m\}$, où c_t^j est le coût (en ressources, temps, argents, etc.) de l'exécution de la tâche t en la durée d_t^j.*

2.2 Contraintes de précédence

Dans le monde r´eel, la possibilit´e d'ex´ecuter une tâche peut d´ependre de plusieurs conditions, comme le temps et les ressources disponibles ou l'ex´ecution d'autres tâches. On considère que les contraintes de pr´ec´edence auparavant sont de la forme d'une relation d'ordre partiel sur un ensemble de tâches T. Dans ce cas, entre deux tâches quelconques il existe une relation de pr´ec´edence ou il n'existe pas de relation qui les relie. On ´etablit une distinction entre deux sortes de contraintes de pr´ec´edence :

1. *Contraintes de Précédence Conjonctives* : le cas où un groupe de tâches ind´ependantes doit être ex´ecut´e pour permettre l'ex´ecution des certaines autres tâches. Si les tâches $t_1, t_2, \ldots, t_k$ doivent toutes être ex´ecut´ees avant l'ex´ecution d'une tâche t, on ´ecrit symboliquement : $[t_1, t_2, \ldots, t_k] \rightarrow t$.
2. *Contraintes de Précédence Disjonctives* : le cas où une seule tâche parmi un groupe de tâches doit être ex´ecut´ee avant que la tâche en question ne puisse être ex´ecut´ee. S'il suffit qu'une tâche

parmi les tâches $t_1, t_2, \ldots, t_k$ s'ex´ecute pour que la tâche t puisse s'ex´ecuter, on ´ecrit symboliquement : $t_1|t_2|\ldots|t_k \rightarrow t$.

À noter que parfois on doit attendre un peu de temps après la fin de l'ex´ecution d'une tâche et avant le d´ebut de l'ex´ecution de la tâche suivante.
On d´esigne ce temps de d´elai entre les tâches t_i et t_j par δ_{t_i,t_j}.
Nous pouvons aussi repr´esenter ce d´elai par une tâche interm´ediaire avec une dur´ee ´egale à ce d´elai d'attente ($\Delta_{t_k} = \delta_{t_i,t_j}$) tel que $t_i \rightarrow t_k \rightarrow t_j$, $c_{t_k} = 0$, $I^-_{t_k} = I^-_{t_i} + min\{\Delta_{t_i}\}$, $I^+_{t_k} = I^+_{t_i} + \delta_{t_i,t_j}$ et $pr_{t_k} = 1$. Cette repr´esentation simplifie le calcul des intervalles temporels d´ecrits dans la section 4 mais complique la recherche des plans dans le graphe puisqu'il y aura plus de tâches à explorer.

2.3 Le graphe de contraintes temporelles

Le graphe de contraintes temporelles associ´e à un agent d´ecrit les contraintes temporelles sur les tâches et les relations de pr´ec´edence entre elles.

Définition 2 *Un graphe de contraintes temporelles est un graphe simple orienté acyclique $G = (T, E)$ où l'on se donne de plus :*

- *pour chaque tâche $t \in T$, la liste $< I^-_t, I^+_t, \Delta_t, Pr_t, C_t >$ des données associées à t ;*
- *un ensemble de contraintes de précédence conjonctives $[t_1, t_2, \ldots, t_k] \rightarrow t$;*
- *un ensemble de contraintes de précédence disjonctives $t_1|t_2|\ldots|t_k \rightarrow t$;*
- *un ensemble de contraintes de délai $\{\delta_{t,t'}, t \in T, t' \in T, t \rightarrow t'\}$*

Il est à noter aussi que les nœuds qui forment l'ensemble T sont divis´es en trois sous-ensembles $T = T_I \cup T_M \cup T_F$ de tâches tels que T_I est l'ensemble de tâches qui n'ont pas de pr´ed´ecesseurs (les tâches initiales), T_M est l'ensemble de tâches qui ont des pr´ed´ecesseurs et des successeurs et T_F est l'ensemble des tâches finales qui n'ont pas de successeurs et qui, en g´en´eral, appartiennent à l'ensemble des buts. La figure 1 repr´esente un graphe de contraintes temporelles tel que $T_I = \{t_1, t_2, t_3, t_4, t_5\}, T_F = \{t_{17}, t_{18}\}$ et $T_M = \{t_6, ..., t_{16}\}$. Dans la section suivante, nous d´ecrirons le problème temporel à r´esoudre.

3 DESCRIPTION DU PROBLÈME

Classiquement un planificateur dispose en entr´ee d'un problème et d'un domaine de planification. Un problème de planification consiste en une description de l'´etat initial et de l'´etat but. Un domaine de planification est d´ecrit par un ensemble d'actions qui vont permettre des transitions entre ´etats logiques.
Le problème à r´esoudre est un problème de planification temporelle, dont les contraintes de pr´ec´edence (ET/OU, d´elai) et les contraintes temporelles ($\Delta_t, [I^-_t, I^+_t]$).
Le but est de trouver un plan des tâches ex´ecutables qui amène d'un ensemble de tâches initiales à un ensemble de buts et qui satisfait toutes les contraintes. Un tel plan est nomm´e *plan admissible*. Nous nous int´eressons à trouver un plan qui a une grande probabilit´e d'être ex´ecut´e et un temps et un coût r´eduits. On d´efinit le problème comme suit :

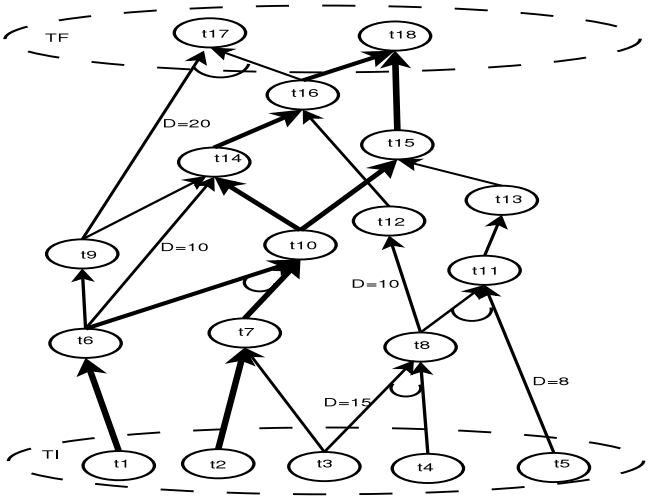

FIG. 1 – Un graphe de contraintes temporelles
Les plans faisables sont marqu´es en gras. Un arc de cercle entre deux arêtes repr´esente une relation conjonctive entre deux tâches.

Définition 3 *Un problème de génération de plan est un couple (G, T') où $G = (T, E)$ est un graphe de contraintes temporelles et où $T' \subset T_F \subset T$ est un sous ensemble de l'ensemble T_F des tâches finales de G.*

Une solution du problème d´ecrit pr´ec´edemment est d´efinie par un ensemble de tâches à ex´ecuter de telle façon que tous les buts demand´es soient atteints. Un plan est un chemin dans le graphe de l'ensemble des tâches initiales T_I à l'ensemble des tâches appartenant à T_F.
Nous utilisons la m´ethode de recherche dans l'espace des plans partiels qui part d'un plan incomplet contenant l'´etat final. Ce plan partiel est ´etendu jusqu'à obtenir un plan complet, appel´e *faisable*, sans d´efauts, c'est à dire qui respecte toutes les contraintes de pr´ec´edence. Ce problème de planification est d´emontr´e NP-complet.
Le plan doit être aussi *admissible* en respectant les contraintes temporelles. On d´esigne par $\mathcal{P}_F$ l'ensemble des plans faisables dans le graphe G et $T(\mathcal{P})$ le sous ensemble de T de toutes les tâches dans un plan faisable $\mathcal{P}$. Par exemple, dans la figure 1, les plans faisables (marqu´es en gras) amenant de $t_2 \in T_I$ à $t_{18} \in T_G$ sont $[[[t_2, t_7], [t_1, t_6]], t_{10}, t_{14}, t_{16}, t_{18}]$ et $[[[t_2, t_7], [t_1, t_6]], t_{10}, t_{15}, t_{18}]$.

4 PROPAGATION DES INTERVALLES D'EXÉCUTION

Étant donn´e un graphe de contraintes temporelles $G = (T, E)$, notre but est de calculer pour chaque tâche des plans faisables l'ensemble des intervalles de temps d'ex´ecution possibles. Nous avons d´evelopp´e un algorithme bas´e sur une propagation temporelle dans le

graphe des tâches. Cette propagation organise le graphe en plusieurs niveaux : l_0 contient les tâches initiales du graphe (T_I), l_1 contient tous les nœuds (tâches) qui succèdent directement aux tâches du niveau l_0. Les feuilles sont les nœuds qui n'ont pas de successeurs (T_F). Pour chaque tâche dans un niveau donné l, nous calculons tous les intervalles temporels possibles à partir de ses prédécesseurs en respectant toutes les contraintes temporelles et les contraintes de précédence.

Nous désignons par st_t la date de début d'exécution, par et_t la date de fin d'exécution, par ST_t l'ensemble de dates de début possibles, par ET_t l'ensemble de dates de fin possibles et par I_t l'ensemble des intervalles temporels possibles pour une tâche t. Tous les intervalles d'exécution possibles de chaque tâche peuvent être calculés hors-ligne (avant l'exécution des tâches) en utilisant un algorithme similaire à celui décrit dans [BW00]. Les dates de début et les dates de fin d'une tâche $t \in T(\mathcal{P})$ en supposant que $\Delta_t = \{d_t^1, d_t^2, ..., d_t^m\}$ et $i = 1...m$ sont calculées comme suit :

- *niveau* l_0 :$t \in T_I$ (t est une tâche initiale). Supposons que l'agent se réveille à la date $start\_time$.
  - $st_t = max\{I_t^-, start\_time\}$, $et_t^i = st_t + d_t^i$
  - $I_t^i = [st_t, et_t^i]$
  - $ST_t = \{st_t\}, ET_t = \{et_t^i\}$ et $I_t = \{I_t^i\}$
- *niveau* l_i : les dates de début possibles de chaque nœud (tâche) appartenant à un niveau l_i, sont calculées à partir des dates de fin des prédécesseurs du nœud. Les dates de début et de fin des tâches autres que la racine sont donc calculées de la manière suivante :
  - si $t \in T_M \cup T_F$ (t est une tâche intermédiaire ou finale) et t est un nœud OR tel que $ET_{t_i} = \{et_{t_i}^1, ..., et_{t_i}^n\}$ (déjà calculée) est l'ensemble des dates de fin possibles de la tâche précédente t_i et supposons que $\Delta_t = \{d_t^1, d_t^2, ..., d_t^m\}$ alors
    - $st_t^j = max(I_t^-, et_{t_i}^j + \delta_{t_i,t})$, $et_t^{jk} = st_t^j + d_t^k$
    - $I_t^{jk} = [st_t^j, et_t^{jk}]$, $I_t = \{I_t^{jk}\}$
    - $ST_t = \{st_t^j\}$ et $ET_t = \{et_t^{jk}\}$ tel que $j = 1..n$ et $k = 1..m$
  - si $t \in T_M \cup T_F$ (t est une tâche intermédiaire ou finale) tel que $\Delta_t = \{d_t^1, d_t^2, ..., d_t^m\}$ et t a n tâches $\{t_1, \ldots, t_n\}$ comme prédécesseurs immédiats (le cas d'un nœud ET) tel que pour chaque tâche t_i $(i = 1..n)$, $ET_{t_i} = \{et_{t_i}^1, et_{t_i}^2, ..., et_{t_i}^{j_i}\}$ alors :
    - $st_t^k = max(I_t^-, max(et_{t_i}^k + \delta_{t_i,t}))$, $et_t^{kr} = st_t^k + d_t^r$
    - $I_t^{kr} = [st_t^k, et_t^{kr}]$, $I_t = \{I_t^{kr}\}$
    - $ST_t = \{st_t^k\}$ et $ET_t = \{et_t^{kr}\}$ tel que $k = 1..j_i$, $r = 1..m$ et $i = 1..n$.

Un plan *faisable* $\mathcal{P} \in \mathcal{P}_F$, pour qu'il devienne *admissible*, doit vérifier toutes les contraintes temporelles. Plus formellement : Pour toute tâche $t_i \in T(\mathcal{P})$ et pour chaque $d_{t_i}^j \in \Delta_{t_i}$, ces conditions doivent être vraies : $st_{t_i}^j \geq I_{t_i}^-$, et $st_{t_i}^j + d_{t_i}^j \leq I_{t_i}^+$.

Nous désignons par $\mathcal{P}_A$ l'ensemble de tous les plans admissibles $\mathcal{P}$ de $\mathcal{P}_F$. $\mathcal{P} = \{t_i/I_{t_i}^j, t_{i+1}/I_{t_{i+1}}^k, ..., t_n/I_{t_n}^l\}$ est un plan formé par un ensemble de tâches $t_i, ..., t_n$ exécutées dans leurs intervalles temporels $I_{t_i}^j, ..., I_{t_n}^l$ respectivement. Nous avons $\mathcal{P}_A = \bigcup_{i=1}^n \mathcal{P}_i$, tel que $\mathcal{P}_i \in \mathcal{P}_F$ est un plan admissible.

5 ALGORITHME DE PROPAGATION DES PROBABILITÉS

Nous décrivons dans cette section comment attribuer une probabilité à chacun des intervalles calculés précédemment. En effet, la probabilité d'un intervalle d'exécution I_t^i dépend de la date à laquelle la tâche commence (date de fin de la tâche précédente) et de la probabilité de sa durée d'exécution $p(\delta_t^i)$ (pour simplifier, nous considérons dans le reste de cet article que les délais entre les tâches sont égaux à zéro). Soit I_t^i un intervalle, st_t^i sa date de début et et_t^i sa date de fin ($I_t^i = [st_t^i, et_t^i]$). La probabilité qu'une tâche t s'exécute dans l'intervalle I_t^i est égale à la probabilité que son exécution commence à st_t^i et se termine à et_t^i, autrement dit que l'exécution de la tâche commence à st_t^i et dure $et_t^i - st_t^i$ unités de temps. Afin de commencer l'exécution d'une tâche t, tous les prédécesseurs de cette dernière doivent avoir été exécutés. La probabilité que l'exécution de t commence à st_t^i est définie par :

- $pr_{start}(st_t^i|et_{t'}^j)$ si t est un nœud OU et il a un seul prédécesseur t' qui termine son exécution à $et_{t'}^j$:
  - si $st_t^i < et_{t'}^j$, $pr_{start}(st_t^i|et_{t'}^j) = 0$
  - si $st_t^i \geq et_{t'}^j$, $pr_{start}(st_t^i|et_{t'}^j) = 1$
- $pr_{start}(st_t^r|et_{t_i}^k)$ si t est un nœud ET et il a plusieurs prédécesseurs $\{t_1, t_2, ..., t_n\}$. Soit $t_i \in \{t_1, t_2, ..., t_n\}$ tel que $ET_{t_i} = \{et_{t_i}^1, et_{t_i}^2, ..., et_{t_i}^{j_i}\}$ où j_i représente le nombre de dates de fin possibles de t_i et $ST_t = \{st_t^1, st_t^2, ..., st_t^m\}$ est l'ensemble de dates de début possibles de t alors :
  - si $st_t^r < max\{et_{t_i}^k\}$ alors $pr_{start}(st_t^r|et_{t_i}^k) = 0$
  - si $st_t^r \geq max\{et_{t_i}^k\}$ alors $pr_{start}(st_t^r|et_{t_i}^k) = 1$

Un cas spécial doit être fait de la première tâche. En effet, les tâches initiales ($\in T_I$) n'ont pas de prédécesseurs. La probabilité qu'une tâche initiale t commence son exécution à st_t est donnée par $pr_{start}(st_t) = 1$. $pr_t(d_t^i|st_t^i)$ désigne la probabilité que l'exécution de t dure d_t^i unités de temps si elle a commencé à st_t. Les probabilités sur les durées dépendent donc des dates de début d'exécution.

La probabilité qu'une tâche s'exécute dans un intervalle I_t^i sachant que son prédécesseur t' a terminé son exécution à $et_{t'}^j$ est notée $Pr_w(I_t^i|et_{t'}^j)$. Elle est définie de la manière suivante :

$$Pr_w(I_t^i|et_{t'}^j) = pr_{start}(st_t^i|et_{t'}^j) * pr_t(d_t^i|st_t^i) \quad (1)$$

où $et_{t'}^j$ est la date de fin de la dernière tâche exécutée (prédécesseur). En effet, une tâche t ne pourra commencer à être exécutée que lorsque son prédécesseur t' aura terminé. La date de début de t dépend donc de la date de fin de t'. Dans le cas où t, tel que $\Delta_t = \{d_t^1, d_t^2, ..., d_t^m\}$ est un nœud ET et a plusieurs prédécesseurs $t_1, t_2, ..., t_n$ tel que $t_i \in \{t_1, t_2, ..., t_n\}$ termine son exécution à $et_{t_i}^{k_{t_i}}$ ($k_{t_i} = 1..j_i$ tel que j_i représente le nombre de dates de fin possibles de t_i), on a :

$$Pr_w(I_t^r|e_{t_i}^{k_{t_i}}) = pr_{start}(s_t^r|e_{t_i}^{k_{t_i}}) * pr_t(d_t^r|s_t^r) \quad (2)$$

où $r = 1..m$.

Un cas sp´ecial doit être envisag´e pour les tâches initiales :

$$Pr_w(I_t^r) = pr_{start}(s_t) * pr_t(d_t^r|s_t) \quad (3)$$

Supposons que la probabilit´e que l'ex´ecution de la tâche t_i dure 2 unit´es de temps lorsqu'elle commence à 5, soit ´egale à 0.7 ($Pr_{t_i}(d_{t_i}^j = 2|st_{t_i}^j = 5) = 0.7$). Supposons ´egalement que la tâche pr´ec´edente se termine à 3. Consid´erons la tâche t, la probabilit´e de l'intervalle $[5, 7]$ est alors 0.7 : $Pr_w(I_{t_i}^j = [5, 7]|et_{t_{i'}}^{j'} = 3) = 1 \times 0.7 = 0.7$
Si par exemple, la tâche $t_{i'}$ se termine à 7, on a : $Pr_w(I_{t_i}^j = [5, 7]|et_{t_{i'}}^{j'} = 7) = 0 \times 0.7 = 0$

6 SÉLECTION ET EXÉCUTION DU MEILLEUR PLAN

Étant donn´e que l'ordre de recherche des plans est arbitraire, il est possible d'obtenir plusieurs plans admissibles. Dans ce cas, plusieurs critères de pr´ef´erences (probabilit´e, temps et coût) sont utilis´es pour analyser et comparer les plans et ensuite en choisir un seul pour être ex´ecut´e. Dans cette section, nous proposons la m´ethode utilis´ee pour choisir "le meilleur plan". Nous nous sommes int´eress´es à trouver un plan qui a une forte probabilit´e d'être ex´ecut´e avec un temps et un coût r´eduits.
Pour chaque tâche dans un plan admissible $\mathcal{P}$, nous calculons les utilit´es esp´er´ees de coût et de temps. Soit la tâche t telle que $\Delta_t = \{d_t^1, d_t^2, ..., d_t^m\}$ est l'ensemble des dur´ees d'ex´ecution possibles de t, $C = \{c_t^1, c_t^2, ..., c_t^m\}$ est l'ensemble des coûts d'ex´ecution de t tel que c_t^k est le coût de l'ex´ecution de la tâche t pendant la dur´ee d_t^k ($k = 1..m$).

1. si t a un seul pr´ed´ecesseur t' dont l'ensemble des dates de fin possibles est $ET_{t'} = \{et_{t'}^1, et_{t'}^2, \ldots, et_{t'}^r\}$, et si la probabilit´e que t s'ex´ecute pendant une dur´ee d_t^k sachant que son pr´ed´ecesseur t' termine son ex´ecution à $et_{t'}^j$ est not´ee $Pr_w(I_t^k|et_{t'}^j)$, alors :
   – l'utilit´e esp´er´ee de coût de t est calcul´ee de la façon suivante :

$$\mu(cout(t|t')) = \sum_{j=1}^{r}\sum_{k=1}^{m} Pr_w(I_t^k|et_{t'}^j) * c_t^k \quad (4)$$

   Pour simplifier, on note l'utilit´e esp´er´ee de coût de t par $\mu(cout(t))$.
   – l'utilit´e esp´er´ee de temps de t est calcul´ee de la façon suivante :

$$\mu(temps(t|t')) = \sum_{j=1}^{r}\sum_{k=1}^{m} Pr_w(I_t^k|e_{t'}^j) * \delta_t^k \quad (5)$$

   Pour simplifier, on note l'utilit´e esp´er´ee de temps de t par $\mu(temps(t))$.
2. si t a un ensemble de pr´ed´ecesseurs directs $t_1, t_2, \ldots, t_n$ alors, soit $ET_{t_i} = \{et_{t_i}^1, et_{t_i}^2, \ldots, et_{t_i}^{r_{t_i}}\}$ l'ensemble des dates de fin possibles de $t_i \in \{t_1, t_2, \ldots, t_n\}$ alors :
   – l'utilit´e esp´er´ee de coût de t est calcul´ee de la façon suivante :

$$\mu(cout(t|t_1, \ldots, t_n)) = \sum_{i=1}^{n} \mu(cout(t|t_i)) \quad (6)$$

   Pour simplifier, on note l'utilit´e esp´er´ee de coût de t par $\mu(cout(t))$.
   – l'utilit´e esp´er´ee de coût de t est calcul´ee de la façon suivante :

$$\mu(temps(t|t_1, \ldots, t_n)) = \sum_{i=1}^{n} \mu(temps(t|t_i)) \quad (7)$$

   Pour simplifier, on note l'utilit´e esp´er´ee de temps de t par $\mu(temps(t))$.

Après avoir calcul´e l'utilit´e esp´er´ee du coût et de temps de chaque tâche, nous calculons l'utilit´e esp´er´ee totale de coût et l'utilit´e esp´er´ee totale de temps de chaque plan comme suit :
– l'utilit´e esp´er´ee totale de coût d'un plan admissible $\mathcal{P}$ est calcul´ee en additionnant toutes les utilit´es esp´er´ees de coût de toutes les tâches de ce plan. Plus formellement :

$$\mu(cout(\mathcal{P})) = \sum_{i=1}^{n} \mu(cout(t_i)) \quad (8)$$

tel que n est le nombre des tâches dans le plan $\mathcal{P}$.
– l'utilit´e esp´er´ee totale de temps d'un plan admissible $\mathcal{P}$ ($\mu(temps(\mathcal{P}))$ est initialis´ee à $max(\mu(temps(t_i)))$ où $t_i \in T_F$) est calcul´ee comme suit :
   – si t est un nœud OU et il a un seul pr´ed´ecesseur t' avec une utilit´e esp´er´ee $\mu(temps(t'))$, alors on ajoute $\mu(temps(t'))$ à l'utilit´e esp´er´ee totale du plan. Plus formellement :

$$\mu(temps(\mathcal{P})) = \mu(temps(\mathcal{P})) + \mu(temps(t')) \quad (9)$$

   – si t est un nœud ET et il a un ensemble de pr´ed´ecesseurs $\{t_1, t_2, \ldots, t_n\}$ alors, on ajoute le maximum de l'utilit´e esp´er´ee des tâches $t_1, t_2, \ldots, t_n$ à l'utilit´e esp´er´ee totale du plan. Plus formellement :

$$\mu(temps(\mathcal{P})) = \mu(temps(\mathcal{P})) + max_{i=1}^{n}\mu(temps(t_i)) \quad (10)$$

Pour calculer l'utilit´e esp´er´ee totale de chaque plan $\mathcal{P} \in \mathcal{P}_A$, nous assignons une valeur r´eelle à chaque plan selon les pr´ef´erences de l'utilisateur. Nous traduisons ces pr´ef´erences en une fonction $\mu(\mathcal{P})$: μ appel´ee la fonction utilit´e. En introduisant des coefficients α et β pour le temps et le coût, nous valorisons l'importance des utilit´es donn´ee par l'utilisateur au temps ou au coût. Nous obtenons un problème d'analyse à multi-critères qui nous permet d'ajuster l'importance relative des diverses utilit´es selon les pr´ef´erences de l'utilisateur. Cette utilit´e est d´ecrite de la façon suivante :

$$\mu(\mathcal{P}) = \alpha * \mu(cout(\mathcal{P})) + \beta * \mu(temps(\mathcal{P})) \quad (11)$$

tels que $\alpha + \beta = 1$ et $P \in \mathcal{P}_A$.

Parmi tous les plans admissibles, nous choisissons pour l'ex´ecution celui qui a l'utilit´e minimale :

$$\mathcal{P}^* = argmin_{\mathcal{P}\in\mathcal{P}_A}\mu(\mathcal{P}) \quad (12)$$

À noter que chaque plan $\mathcal{P} \in \mathcal{P}_A$ est form´e de plusieurs ordonnancements en prenant les diff´erents intervalles d'ex´ecution possibles $I^j_{t_i}$ pour chaque tâche t_i de $T(\mathcal{P})$. Plus formellement : $\mathcal{P} = \bigcup \mathcal{P}^f$
tel que $\mathcal{P}^f = \{t_i/I^j_{t_i}, t_{i+1}/I^k_{t_{i+1}}, ..., t_n/I^l_{t_n}\}$
avec $j = 1...m$ où m est le nombre d'intervalles d'ex´ecution possibles de la tâche t_i, $k = 1...s$ où s est le nombre d'intervalles d'ex´ecution possibles de la tâche t_{i+1} et $l = 1...o$ où o est le nombre d'intervalles d'ex´ecution possibles de la tâche t_n.

Le nombre total des ordonnancements d'un plan $\mathcal{P}$ est ´egal au produit des cardinalit´es des ensembles d'intervalles d'ex´ecution possibles de toutes les tâches de $T(\mathcal{P})$. Plus formellement ce nombre est ´egal à $m*s*...*o$.

Pour chaque ordonnancement $\mathcal{P}^*_i = \{t_i/I^j_{t_i}, t_{i+1}/I^k_{t_{i+1}}, ..., t_n/I^l_{t_n}\}$ d'un plan admissible $\mathcal{P}^*$ choisi pour être ex´ecuter, nous calculons sa fonction d'utilit´e, tel que $\alpha + \beta = 1$ et $t_i \in T(P^*)$, d´efini par :

$$\mu(\mathcal{P}^*_i) = \alpha * \mu(cout^*_{t_i}) + \beta * \mu(temps^*_{t_i}) \quad (13)$$

tel que :

$$\mu(cout^*_{t_i}) =$$

$$Pr_w(I^j_{t_i}) * cout(I^j_{t_i}) + Pr_w(I^k_{t_{i+1}}|et^j_{t_i}) * cout(I^k_{t_{i+1}}|et^j_{t_i})$$
$$+... + Pr_w(I^l_{t_n}|et^s_{t_{n-1}}) * cout(I^l_{t_n}|et^s_{t_{n-1}})$$

et

$$\mu(temps^*_{t_i}) =$$

$$Pr_w(I^j_{t_i}) * temps(I^j_{t_i}) + Pr_w(I^k_{t_{i+1}}|et^j_{t_i}) * temps(I^k_{t_{i+1}}|et^j_{t_i})$$
$$+... + Pr_w(I^l_{t_n}|et^s_{t_{n-1}}) * temps(I^l_{t_n}|et^s_{t_{n-1}})$$

De tous les ordonnancements form´es par $\mathcal{P}^*$ nous choisissons celui qui a l'utilit´e minimal :

$$\mathcal{P}^*_{exec} = argmin_{\mathcal{P}^*_i\in P^*}\mu(\mathcal{P}^*_i) \quad (14)$$

Cet ordonnancement d´etermine les intervalles d'ex´ecution esp´er´es, pour chaque tâche, d'être ex´ecut´es par l'agent.

7 ANALYSE ET DISCUSSION

Dans cette section, nous analysons le nombre d'intervalles temporels pour chaque tâche t_i dans le graphe $G = (T, E)$. Dans le pire des cas, le nombre d'intervalles temporels pour une tâche initiale (dans T_I) est ´egale à la cardinalit´e de son ensemble de dur´ees. Le nombre d'intervalles temporels pour une tâche interm´ediaire ou finale (dans T_M ou T_F) de la forme d'un nœud OU est ´egale à la cardinalit´e de son ensemble de dur´ees multipli´ee par la cardinalit´e de l'ensemble de dates de fins de la tâche pr´ec´edente. Le nombre d'intervalles temporels pour une tâche interm´ediaire ou finale (dans T_M ou T_F) de la forme d'un nœud ET est ´egal à la cardinalit´e de son ensemble de dur´ees multipli´ee par toutes les cardinalit´es de tous les ensembles de date de fin des tâches pr´ec´edentes. Plus formellement, dans le pire de cas, nous avons :

- si $t \in T_I$ alors $|I_t| = |\Delta_t|$,
- si $t \in T_M \cup T_F$ et t est un nœud OU alors $|I_t| = |\Delta_t| * |ET_{t'}|$ tel que t' est une tâche directement pr´ec´edente de la tâche t,
- si $t \in T_M \cup T_F$ est un nœud ET tel que t a un ensemble de pr´ed´ecesseurs directs $\{t_1, t_2, \ldots, t_n\}$ alors $|I_t| = |\Delta_t| * \prod_{i=1}^{n} |E_{t_i}|$

Le nombre d'intervalles temporels croit exponentiellement avec la taille du plan dans le graphe. Mais il est possible qu'on obtienne plusieurs intervalles temporels identiques (qui ont les mêmes dates de d´ebut et de fin), mais avec des probabilit´es d'ex´ecution diff´erentes. Dans ce cas, nous pouvons consid´erer que ces intervalles temporels forment un seul intervalle avec la date de d´ebut et de fin et avec une probabilit´e ´egale à la somme de toutes les probabilit´es de tous les intervalles identiques (d´emonstration dans la section 7.1). Par exemple, si pour la tâche t_i nous obtenons les cinq intervalles temporels $[6, 7], [6, 8], [7, 9], [6, 7]$, et $[7, 9]$ avec les probabilit´es $0.10, 0.20, 0.09, 0.21, 0.40$ respectivement, on peut dire qu'on a seulement trois intervalles temporels $[6, 7], [6, 8]$ et $[7, 9]$ avec les probabilit´es $0.31, 0.20$ et 0.49 respectivement. A noter que $\sum pr^j_{t_i} \leq 1$.

7.1 Réduction de nombre d'intervalles

Dans cette section, nous analysons les diff´erents cas de calcul des intervalles temporels des tâches afin de le r´eduire.

- niveau l_0 : $t \in T_I$ (t est une tâche initiale) tel que $\Delta_t = \{d^1_t, d^2_t, ..., d^m_t\}$:
  Le nombre d'intervalles de t est ´egal au nombre de dur´ees qui satisfont la condition : $max(start\_time, I^-_t) + d^i_t \leq I^+_t$, où $[I^-_t, I^+_t]$ est la fenêtre temporelle de la tâche t et $d^i_t \in \Delta_t$.
- niveau l_i : $t \in T_M \cup T_F$ (t est une tâche interm´ediaire ou finale) :
  Commençons par le cas où t est un nœud OU et a une seule tâche directement pr´ec´edente t' ($t' \rightarrow t$). Supposons que l'ensemble des dates de fins possibles de t' est $ET_{t'} = \{et^1_{t'}, \ldots, et^p_{t'}\}$, la fenêtre temporelle de la tâche t est $[I^-_t, I^+_t]$ et $\Delta_t = \{d^1_t, d^2_t, ..., d^m_t\}$. On considère la tâche t. Nous distinguons deux cas :
  1. si le maximum des dates de fin $et^j_{t'} \in ET_{t'}$ est inf´erieur ou ´egal à I_t, alors $st_t = I^-_t$ et le nombre d'intervalles temporels possibles de la tâche t est ´egal au nombre de ses dur´ees qui satisfont la condition : $I^-_t + d^i_t \leq I^+_t$.
  2. si le minimum des dates de fin $et^j_{t'} \in ET_{t'}$ est sup´erieur ou ´egal à I_t, alors le nombre d'intervalles temporels possibles de la tâche t est ´egal à la cardinalit´e de $ET_{t'}$ multipli´ee par le nombre de dur´ees de la tâche t qui satisfont la condition : $max(et^j_{t'}, I^-_t) + d^i_t \leq I^+_t$.
  3. sinon, le nombre d'intervalles temporels pos-

sibles de la tâche t est ´egale à un plus le nombre d'intervalles temporels possibles de la tâche t' qui ont diff´erentes valeurs de $et^j_{t'}$ et pour lesquels $et^j_{t'} > I^-_t$ multipli´e par le nombre de dur´ees de la tâche t qui satisfont la condition : $max(et^j_{t'}, I^-_t) + d^i_t \leq I^+_t$.

Nous consid´erons maintenant le cas où t est un nœud ET et a plusieurs tâches directement pr´ec´edentes $\{t_1, t_2, \ldots, t_n\}$. Soit $t_i \in \{t_1, t_2, \ldots, t_n\}$ tel que $E_{t_i} = \{e^1_{t_i}, e^2_{t_i}, \ldots, e^p_{t_i}\}$ est l'ensemble de dates de fin possibles de t_i. Pour calculer l'ensemble de dates de fin possibles de l'ensemble de pr´ed´ecesseurs de t, not´e $ET(t_1, \ldots, t_n)$, on fait une combinaison entre tous les ensembles E_{t_i} pour choisir la date de fin maximale de chaque combinaison. Plus formellement :

$$E(t_1, \ldots, t_n) = \bigcup \{max^p_{k_{t_i}=1}(e^{k_{t_1}}_{t_1}, e^{k_{t_2}}_{t_2}, \ldots, e^{k_{t_n}}_{t_n})\}$$

Par exemple, si une tâche t a trois pr´ed´ecesseurs directs t_1, t_2 et t_3 qui terminent leurs ex´ecutions à $E_{t_1} = \{e^1_{t_1}, e^2_{t_1}, e^3_{t_1}\}$, $E_{t_2} = \{e^1_{t_2}, e^2_{t_2}, e^3_{t_2}\}$ et $E_{t_3} = \{e^1_{t_3}, e^2_{t_3}\}$ respectivement. L'ensemble $E(t_1, t_2, t_3)$ est compos´e de l'ensemble de dates de fin possibles suivant : $\{max(e^1_{t_1}, e^1_{t_2}, e^1_{t_3}), max(e^1_{t_1}, e^1_{t_2}, e^2_{t_3}), etc\}$.

Dans le pire de cas, on a :

$$|E(t_1, \ldots, t_n)| = |E_{t_1}| * |E_{t_2}| * \ldots * |E_{t_n}|$$

Après avoir calcul´e l'ensemble de dates de fin possibles ($E(t_1, \ldots, t_n)$) de l'ensemble de pr´ed´ecesseurs de t, on peut calculer le nombre d'intervalles temporels possibles de t comme d´ecrit ci-dessus.

Dans la section suivante, nous donnons un exemple explicatif pour bien comprendre ces analyses.

7.2 Exemple sur la réduction de nombre d'intervalles

Dans cette section, nous expliquons les analyses pr´ec´edentes grâce à un exemple. Supposons qu'on a trois tâches t_i, t_j et t_k avec ces donn´ees temporelles : $(t_i, [2, 6], \{2, 3\}, \{0.6, 0.4\}, \{10, 20\})$, $(t_j, [2, 8], \{1, 4, 6\}, \{0.1, 0.4, 0.5\}, \{10, 30, 35\})$ et $(t_k, [5, 9], \{1, 2, 4\}, \{0.1, 0.6, 0.3\}, \{15, 30, 35\})$.

Nous repr´esentons ces tâches dans la figure 2 et nous supposons qu'il n'y a pas de d´elais entre les tâches.

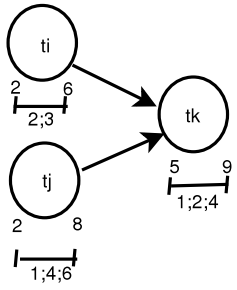

FIG. 2 – Un exemple d'un graphe de contraintes temporelles compos´e de 3 tâches.

Au d´ebut, supposons que t_k est un nœud OU. Consid´erons le plan (t_i, t_k).

Grâce à la propagation temporelle d´ecrite dans le paragraphe 4, nous obtenons : $I^1_{t_i} = [2, 4]$ avec une probabilit´e ´egale à 0.60 et $I^2_{t_i} = [2, 5]$ avec une probabilit´e ´egale à 0.40. Ces deux intervalles temporels satisfont la condition : $et^m_{t_i} = \{4, 5\} \leq I^+_{t_i} = 6$. Pour calculer les intervalles temporels de la tâche t_k, nous utilisons les dates de fins possibles de la tâche t_i. Nous obtenons six intervalles possibles : $[5, 6], [5, 7], [5, 9], [5, 6], [5, 7]$ et $[5, 9]$ avec les probabilit´es $0.06, 0.36, 0.18, 0.04, 0.24$ et 0.12 respectivement. A noter que trois intervalles temporels sont rejet´es. Nous les r´eduisons en un seul intervalle temporel et nous additionnons leurs probabilit´es. Nous obtenons trois intervalles temporels $[5, 6], [5, 7]$ et $[5, 9]$ avec les probabilit´es $0.10, 0.60$ et 0.30 respectivement.

Maintenant supposons que t_k est un nœud ET. Consid´erons le plan $([t_i, t_j], t_k)$. Grâce à la propagation temporelle d´ecrite dans 4, nous obtenons : $I^1_{t_i} = [2, 4]$ avec une probabilit´e ´egale à 0.60 et $I^2_{t_i} = [2, 5]$ avec une probabilit´e ´egale à 0.40. Ces deux intervalles temporels satisfont la condition $et^m_{t_i} = \{4, 5\} \leq I^+_{t_i} = 6$. $I^1_{t_j} = [2, 3]$ avec une probabilit´e ´egale à 0.10, $I^2_{t_j} = [2, 6]$ avec une probabilit´e ´egale à 0.40 et $I^3_{t_j} = [2, 8]$ avec une probabilit´e ´egale à 0.50. Ces intervalles temporels satisfont la condition $et^n_{t_j} = \{3, 6, 8\} \leq I^+_{t_i} = 8$.

Pour calculer les intervalles temporels de la tâche t_k nous utilisons les dates de fins possibles des tâches t_i et t_j. Nous obtenons huit intervalles possibles :
$[5, 6]$, $[5, 7]$, $[5, 8]$, $[6, 7]$, $[6, 8]$, $[6, 10]$, $[8, 9]$, $[8, 10]$, $[8, 12]$, $[5, 6]$, $[5, 7]$, $[5, 9]$, $[6, 7]$, $[6, 8]$, $[6, 10]$, $[8, 9]$, $[8, 10]$ et $[8, 12]$. Remarquons que les intervalles $[6, 10], [8, 10]$ et $[8, 12]$ ne sont pas valides puisque ils ne satisfont pas la condition $et^p_{t_k} = \{10, 12\} \leq I^+_{t_k} = 9$. Nous ´eliminons les plans qui contiennent ces intervalles. Parmi tous les autres intervalles, six intervalles temporels sont r´ep´et´es. Nous les r´eduisons en un seul et nous additionnons leurs probabilit´es. Nous obtenons seulement sept intervalles temporels $[5, 6]$, $[5, 7]$, $[5, 9]$, $[6, 7]$, $[6, 8]$ et $[8, 9]$.

8 EXPÉRIENCES

Le nombre d'intervalles temporels de chaque tâche t_i d´epend du nombre des tâches pr´ed´ecesseurs, de la largeur de sa fenêtre temporelle $[I^-_{t_i}, I^+_{t_i}]$ et de la cardinalit´e de son ensemble de dur´ees $|\Delta_i|$. Quand une de ses contraintes croît, le nombre d'intervalles temporels croît. Le nombre de plans dans le graphe d´epend du nombre d'intervalles temporels des tâches, du nombre de tâches et du nombre de contraintes de pr´ec´edence. Ce nombre croît avec le nombre d'intervalles temporels des tâches et du nombre de tâches et diminue avec le nombre de contraintes de pr´ec´edence. Nous avons r´ealis´e des tests afin de v´erifier l'efficacit´e de notre approche. Ces tests sont faits sur des graphes binaires. Nous avons calcul´e le nombre total de plans qui respectent les contraintes de pr´ec´edence et le nombre total de plans admissibles obtenu après la v´erification de la validit´e des contraintes temporelles. Nous avons fix´e le nombre d'intervalles (5 inter-

valles) pour toutes les tâches puis nous avons augment´e le nombre de tâches. Nous avons commenc´e par un ensemble de 2 tâches (avec 5 intervalles chacune) puis nous avons ajout´e un successeur à chaque tâche n'ayant qu'un seul successeur. Lorsque toutes les tâches ont 0 ou 2 successeurs, nous ajoutons un successeur à chaque feuille. La figure 3 montre le r´esultat obtenu, la ligne continue repr´esente le nombre de plans faisable et la ligne pointill´ee repr´esente le nombre de plans admissibles.

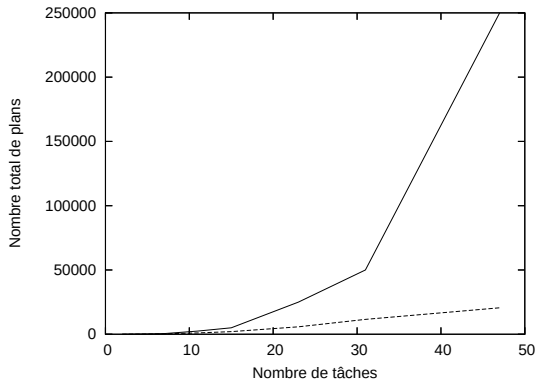

FIG. 3 – Nombre de plans obtenu vs. nombre de plans total (Nb d'intervalles =5)

9 CONCLUSION

Nous avons pr´esent´e une nouvelle approche pour la planification temporelle sous incertitude. Les tâches possèdent des contraintes temporelles, des coûts et des probabilit´es. Nous calculons les plans amenant d'une tâche initiale à une tâche finale en respectant les contraintes de pr´ec´edence et grâce à une propagation temporelle à travers le graphe nous calculons les intervalles d'ex´ecution respectant les contraintes temporelles. Puis nous choisissons un seul plan qui a une forte probabilit´e d'être ex´ecut´e avec un temps et un coût r´eduits.
La suite de notre travail va consister à ´etendre cette approche pour qu'elle soit capable de re-planifier en-ligne dans le cas où un plan ex´ecutable ´echoue.

BIBLIOGRAPHIE

[BDM+02] J. L. Bresina, R. Dearden, N. Meuleau, S. Ramakrishnan, D. E. Smith, and R. Washington. Planning Under Continuous Time and Resource Uncertainty : A Challenge for AI. In *AIPS Workshop on Planning for Temporal Domains*, pages 91–97, April 2002.

[BW00] J. Bresina and R. Washington. Expected Utility Distributions for Flexible Contingent Execution. In *Proceedings of the AAAI-2000 Workshop : Representation Issues for Real-World Planning Systems*, 2000.

[DHW94] D. Draper, S. Hanks, and D. Weld. Probabilistic Planning with Information Gathering and Contingent Execution. In *Proceedings of the Second International Conference on AI Planning Systems, AIPS-94*, pages 31–63, June 1994.

[DKS02] J. Dix, S. Kraus, and V. S. Subrahmanian. Agents dealing with Time and Uncertainty. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS*, pages 912–919. ACM, July 2002.

[DMP91] R. Dechter, I. Meiri, and J. Pearl. Temporal Constraint Network. *Artificial Intelligence*, 49(1-3) :61–95, 1991.

[FN71] R. Fikes and N. J. Nilsson. STRIPS : A New Approach to the Application of Theorem Proving to Problem Solving. *Artif. Intell.*, 2(3/4) :189–208, 1971.

[GL94] M. Ghallab and H. Laruelle. Representation and Control in IxTeT, a Temporal Planner. In *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems, AIPS-94*, pages 61–67, June 1994.

[KHW94] N. Kushmerick, S. Hanks, and D. Weld. An Algorithm for Probabilistic Least-Commitment Planning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence,AAAI-94*, pages 1073–1078, August 1994.

[LI04] S. Lemai and F. Ingrand. Interleaving Temporal Planning and Execution in Robotics Domains. In *AAAI*, pages 617–622, 2004.

[MMV01] P. Morris, N. Muscettola, and T. Vidal. Dynamic Control of Plans with Temporal Uncertainty. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI-01*, pages 494–502. Morgan Kaufmann, 2001.

[PG02] J. C. Pemberton and L. G. Greenwald. On the Need for Dynamic Scheduling of Imagings Satellites. In *Proceedings of the American Society for Photogrammetry and Remote Sensing, ASPRS-02*, 2002.

[TVP03] I. Tsamardinos, T. Vidal, and M. E. Pollack. CTP : A New Constrained-based Formalism for Conditional Temporal Planning. *Constraints journal, special issue on Planning*, 8(4) :365–388, October 2003.