

Approche et cadre de modélisation pour l'évaluation de l'impact de perturbations sur un ordonnancement

Sara Himmiche^{1,2}, Pascale Marangé^{1,2}, Alexis Aubry^{1,2}, and Jean-François Pétin^{1,2}

¹ Université de Lorraine, CRAN, UMR 7039, F-54506 Vandœuvre-lès-Nancy

² CNRS, CRAN, UMR 7039, France

{sara.himmiche, pascale.marange, alexis.aubry, jean-francois.petin}@univ-lorraine.fr

Abstract

L'ordonnancement de la production est un problème majeur aussi bien dans le monde industriel que celui de la recherche théorique. Ce problème est généralement traité dans le but de proposer des ordonnancements optimaux. Dans le cadre d'un environnement de production dynamique, la génération classique d'ordonnements n'est plus suffisante. La question importante qui se pose aujourd'hui est: Comment prendre en compte les contraintes de flexibilité et d'agilité qui caractérisent ces systèmes et leurs perturbations, dans le choix de l'ordonnement à exécuter ?

Cet article répond à cette question en proposant une approche générique pour évaluer la robustesse d'un ordonnancement face aux perturbations. L'approche proposée est basée sur des modèles de systèmes à événements discrets stochastiques et une méthode d'évaluation par model-checking, dans le but de définir une approche et un cadre de modélisation modulaire. Plus précisément, les automates temporisés stochastiques sont utilisés dans le but de modéliser le problème d'ordonnement sous perturbations et le model-checking statistique est utilisé pour évaluer ces modèles et ainsi permettre le calcul d'un indicateur de robustesse qui sera transmis au décideur.

1 Introduction

La question de l'ordonnement de la production est un problème de décision important dans le secteur industriel. En effet, pour garantir les meilleures performances de production, le décideur doit trouver un ordonnancement adapté à son système de production et aux contraintes associées. Un problème d'ordonnement de la production consiste généralement à (i) affecter les ressources de l'atelier aux opérations, nécessaires à la fabrication de jobs, (ii) définir l'ordre de réalisation des opérations sur les ressources et (iii) éventuellement définir la date de début et de fin de chaque opération. L'ordonnement obtenu doit répondre aux contraintes de la gamme logique du produit et de l'atelier (contraintes de précédence, non préemption des opérations, etc.) et aux objectifs de production.

Ce problème est généralement traité dans le but d'obtenir des solutions d'ordonnement optimales notamment dans le domaine de la Recherche Opérationnelle. Cependant, la plupart de ces travaux considèrent que l'environnement de production est stable et que les données de production sont connues et certaines. Ce qui peut être considéré comme une hypothèse forte dans le cas des systèmes de production, car en pratique, un système de production évolue dans un environnement dynamique et doit faire face à de nombreuses perturbations qui peuvent survenir pendant la production.

Dans le milieu de la Recherche Opérationnelle, la question de l'ordonnement sous perturbations a été abordée en profondeur [5] [13]. L'objectif de ces travaux consiste à construire des ordonnancements robustes qui garantissent certaines performances malgré la présence des

perturbations. L'inconvénient de ces approches est qu'elles sont dédiées à un type de problème d'ordonnement donné. Au cours des dernières années, des approches alternatives, dans le domaine des Systèmes à Événements Discrets (SED), ont émergé pour modéliser et résoudre des problèmes d'ordonnement déterministes. Les premiers résultats montrent que les approches sont adaptables aux problèmes d'atelier considérés. De plus leurs caractéristiques en font des candidats prometteurs afin de prendre en compte des perturbations [21, 7, 12, 9, 18].

L'objectif de ce papier est de proposer une approche d'évaluation de la robustesse d'ordonnements soumis à des perturbations. Cette approche repose sur i) une modélisation par automates temporisés stochastiques (ATS) à partir de patrons instanciables, ii) une vérification par model-checking d'une propriété de robustesse.

Deux questions de recherche sont abordées dans ce papier : Comment développer des modèles adaptables et agiles basés sur des (ATS) pour les problèmes d'ordonnement sous perturbations ? Comment utiliser ces modèles et les outils associés afin d'évaluer le niveau de robustesse d'un ordonnancement dans un contexte perturbé ?

Ce papier est structuré comme suit. La section 2 présente le contexte du problème. La section 3 est consacrée à la présentation générale de l'approche proposée et des hypothèses connexes. La section 4 se concentre sur la modélisation basée sur les automates temporisés stochastiques. La section 5 se concentre sur la formalisation et l'évaluation de la robustesse à l'aide du model checking. La section 6 présente une application de l'approche sur une étude de cas. Quelques conclusions et perspectives sont présentées à la fin de ce papier.

2 Contexte: Ordonnement sous perturbations et robustesse

Un atelier est généralement défini par un ensemble J de jobs j (NbJ) qui doivent être traités par un ensemble R de ressources r (NbR). Chaque job j est défini par une gamme, regroupant les opérations o_{jk} à réaliser dans un ensemble O_j^J (o_{jk} est la $k^{ième}$ opération dans la gamme de j). Une durée d'exécution d_{jkr} est associée à l'exécution de l'opération o_{jk} sur la ressource r . Dans ce papier, les ressources peuvent subir des défaillances (mais être réparées avec une durée d_{rep}). Les ateliers de production sont de natures différentes et peuvent engendrer des contraintes sur l'ordre d'exécution des opérations. Par exemple, le Flow Shop et le Job Shop ont des contraintes de précédence sur les opérations d'un job (définies par sa gamme). Alors que pour l'Open Shop, les opérations d'un job peuvent être exécutées dans n'importe quel ordre. Dans un ordonnancement déterministe S_i , les paramètres et les informations sont considérés comme certains. A partir de l'ordonnement déterministe, le séquençement des opérations sur la ressource génère des contraintes supplémentaires à prendre en compte lors de l'évaluation, ainsi que le temps d'exécution (appelé également Makespan) $C_{max}^{ref}(S_i)$.

En pratique, des perturbations peuvent remettre en cause cet ordonnancement. En effet, deux types de perturbations sont recensées : i) Incertitudes et ii) Aléas. Les incertitudes sont définies comme étant la différence entre les informations prévues et les informations réelles disponibles (incertitudes sur le volume de la demande, durée de l'opération, etc.). Les aléas sont définis par l'occurrence d'événements incontrôlables (panne de machine, commande urgente, etc.). Pour intégrer les perturbations dans le problème d'ordonnement, différents modèles existent dans la littérature [17] : description bornées ou par scénario, description probabiliste ou modélisation stochastique et description floue. Dans cet article, nous utiliserons la description probabiliste en considérant les perturbations comme des variables aléatoires suivant une distribution de probabilité. En effet, dans le cadre de l'industrie 4.0, les données historiques

recueillies à partir de composants de l'atelier peuvent être utilisées pour identifier la distribution de probabilité d'un paramètre perturbé. Cette identification n'entre pas dans le cadre de cet article et les distributions de probabilités seront considérées parfaitement connues.

Face à ces perturbations, une approche consiste à construire des solutions d'ordonnement robustes ce qui pose la question de l'évaluation de cette robustesse. Afin d'évaluer l'impact des perturbations, la notion de robustesse a été définie comme suit: un ordonnancement robuste doit maintenir certaines performances malgré les perturbations et variations générées par le système de production ou son environnement [5]. Pour évaluer la robustesse, une métrique doit être définie. Dans ce cadre, nous utilisons la métrique définie dans [6] qui permet d'évaluer le niveau de robustesse (ou niveau de service) comme : *“la probabilité qu'un critère soit inférieur (resp. supérieur) ou égal à une valeur donnée ”*. Par exemple, si dans le problème d'ordonnement, le critère considéré est le Makespan, le niveau de robustesse est la probabilité que le celui-ci soit inférieur ou égal à une deadline donnée définie par le décideur. Formellement, cette métrique est donnée par l'équation (1), avec Pr la probabilité que le makespan C_{max} de l'ordonnement S_i soumis à un ensemble de perturbations $Pert$ soit inférieur ou égal à une deadline donnée \tilde{d} .

$$RL(S_i, Pert, \tilde{d}) = Pr(C_{max}(S_i, Pert) \leq \tilde{d}) \quad (1)$$

Dans la communauté de la Recherche Opérationnelle, [13] et [5] présentent des approches pour générer des ordonnancements robustes sous perturbations (incertitude sur la durée ou la demande, panne machine). Même si les techniques basées sur la recherche opérationnelle s'avèrent performantes et efficaces pour l'ordonnement sous perturbations, l'inconvénient principal est que ces approches et leurs modèles sont dédiées à des problèmes d'ordonnement donnés (liés à l'atelier et à la perturbation considérée) ce qui les rends peu évolutifs et non réutilisables pour d'autres problèmes.

Dans le domaine de la sûreté et de la sécurité, les SED sont généralement utilisés pour évaluer la fiabilité, la disponibilité, la maintenabilité et le diagnostique. Ainsi, ils ont prouvé leur efficacité dans la modélisation des systèmes stochastiques et dynamiques. Grâce à ces atouts, les SED peuvent être considérés comme de bons candidats pour résoudre le problème d'ordonnement en général, comme dans [20], [18]. De plus, il existe plusieurs travaux qui explorent l'utilisation des SED stochastiques pour l'ordonnement sous perturbations [1]. A ce titre, [15] proposent une modélisation stochastique basée sur un langage de file d'attente pour évaluer l'impact d'une perturbation sur les temps d'attente des opérations. Dans [16], les réseaux de pétri temporisés sont utilisés afin de tenir compte du risque de déviation d'un ordonnancement en présence d'événements incontrôlables. Dans nos précédents travaux [10] [11], nous proposons une approche d'évaluation de la robustesse spécifiée pour les incertitudes sur les durées d'exécution des opérations ainsi que pour les prise en compte des pannes machines en utilisant les automates temporisés stochastiques et avons démontré la faisabilité de l'approche. L'objectif est ici d'aller plus loin et de généraliser l'approche à tout type de perturbations.

L'hypothèse de recherche de cet article est que les modèles et outils des SED stochastiques peuvent être utilisés pour évaluer la robustesse d'une solution d'ordonnements sous plusieurs perturbations et indépendamment du type de l'atelier. En effet, nous proposons une approche générique qui permettra i) de prendre en compte des perturbations de différents types et ii) d'évaluer la robustesse d'un ensemble d'ordonnements soumis à ces perturbations.

3 Processus d'évaluation de la robustesse - vue d'ensemble

L'objectif principal du processus d'évaluation est d'aider le décideur à choisir l'ordonnement "robuste" qui répond aux contraintes de son atelier ainsi que la deadline à satisfaire. Les

données d'entrée fournies par le décideur sont, un ensemble d'ordonnancements déterministes S_i à évaluer ainsi que le makespan de chaque ordonnancement $C_{max}^{ref}(S_i)$, l'ensemble de perturbations $Pert$ à considérer et la deadline \tilde{d} à satisfaire.

Le processus d'évaluation proposé est basé sur deux étapes essentielles. La première consiste à modéliser le comportement des ordonnancements et celui des perturbations. Cette étape est importante pour représenter les caractéristiques du problème ainsi que les différentes interactions qui existent entre ces éléments.

La deuxième étape est l'évaluation de la robustesse. L'objectif de cette étape est la formulation de la mesure de robustesse donnée dans l'équation (1). La sortie du processus d'évaluation est alors l'ensemble des niveaux de robustesse $RL_i(S_i, Pert, \tilde{d})$ qui sont associés à chaque ordonnancement et aux perturbations considérées. ces niveaux de robustesse sont remis au décideur afin qu'il puisse choisir l'ordonnancement à exécuter dans l'atelier.

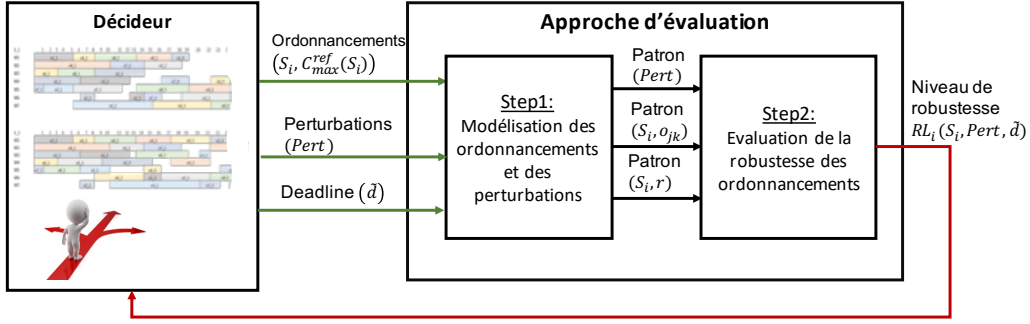


Figure 1: Processus d'évaluation de la robustesse

4 Étape 1 : modélisation du problème

Pour modéliser le comportement des ordonnancements et des perturbations, les modèles doivent tenir compte des caractéristiques dynamiques telles que la communication entre les éléments de l'atelier, le temps d'exécution et le comportement probabiliste des perturbations. Le langage choisi dans cet article est celui des automates temporisés stochastiques (ATS) (1). En effet, il s'agit d'une extension des automates temporisés bien connus [2] qui s'enrichit par le partage de variables, les événements synchronisant et les caractéristiques probabilistes [14].

Définition 1. *Formellement, un automate stochastique temporisé est présenté comme le n-uplet $A = (L, V, E, C, Inv, P, T, L_m, l_0, v_0)$ avec :*

- L est un ensemble fini de localités (notées par des sommets).
- V est un ensemble fini de variables.
- E est un ensemble fini d'événements synchronisant avec $E = E_u \cup E_{\bar{u}}$.
 - E_u est un ensemble fini d'événements urgents. Pour éviter que l'automate reste dans une localité, alors qu'une transition peut être franchie, un événement est déclaré urgent.
 - $E_{\bar{u}}$ est un ensemble fini d'événements non urgents.
- C est un ensemble fini d'horloges.
- Inv est un ensemble d'invariants (notés dans les localités).

- P est un ensemble de probabilités: *i*) discrète sur l'ensemble des transitions (à partir d'une localité, une transition peut conduire dans différentes localités l_i avec des probabilités p_i où $\sum p_i = 1$ (noté par des arcs en pointillés). *ii*) continue sur des variables d'automates (la condition de franchissement d'une transition est définie de façon aléatoire par une distribution de probabilités).
- T est un ensemble fini de transitions (notées par des arcs) $(l, e, g, m, l') \in (L \times E \times G \times M \times L)$ où l et l' sont respectivement les deux localités début et arrivée. Sur une transition trois éléments sont définis: *i*) Une garde (condition sur les variables, notées entre crochets $[]$) $g \in G$, *ii*) Une mise à jour (sur les variables ou les horloges, notée sous la transition) $m \in M$, *iii*) L'événement de synchronisation $e \in E$ (noté en dessous des gardes).
- $L_m \subseteq L$ est l'ensemble des localités marquées (notées par un double sommet).
- $l_0 \in L$ est la localité initial de l'automate.
- v_0 est le vecteur d'initialisation des variables.

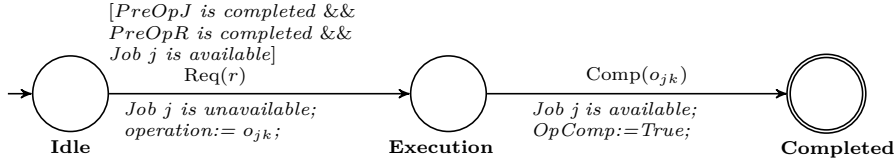
Pour modéliser à la fois les ordonnancements et les perturbations, nous proposons une approche modulaire basée sur plusieurs patrons. D'une part, l'ordonnancement est modélisé avec deux patrons, le premier présentant le comportement de l'opération à exécuter et le deuxième, le comportement de la ressource qui exécute l'opération. D'autre part, les perturbations sont représentées par deux patrons qui modélisent soit le comportement d'une incertitude soit d'un aléa. Chaque modèle doit être instancié en fonction des caractéristiques de l'atelier, du nombre d'opérations et de ressources, des caractéristiques et du nombre de perturbations.

4.1 Modélisation de d'ordonnement

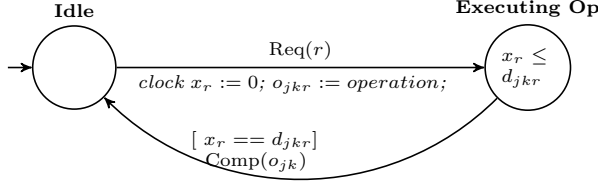
L'exécution d'un ordonnancement est modélisée par deux patrons (opération, ressource) échangeant des informations.

Le patron d'opération (fig.2a) modélise le comportement d'une opération o_{jk} . A partir de la localité **Idle**, lorsque la garde prenant en compte les contraintes de précédence de la gamme du job, les contraintes de la séquence sur la ressource et la disponibilité du job ($[PreOpJ \text{ is completed} \ \&\& \ PreOpR \ \text{is completed} \ \&\& \ Job \ j \ \text{is available}]$) est satisfaite, la transition est franchie vers la localité **Execution**. La vérification de la disponibilité du job permet de garantir le non chevauchement de plusieurs opérations d'un même job et une exécution sûre de l'ordonnancement déterministe donné. De plus lorsque la garde est satisfaite, l'opération o_{jk} se synchronise avec la ressource r qui lui est allouée en utilisant l'événement de requête ($Req(r)$). La disponibilité du job est alors mise à jour et l'identifiant de l'opération est sauvegardé ($opération := o_{jk}$), pour garder en mémoire l'opération en cours d'exécution. Dans la localité **Execution**, le modèle *opération* attend l'occurrence de l'événement de synchronisation ($Comp(o_{jk})$), indiquant la fin de l'exécution de l'opération et franchit la transition vers sa localité marquée **Completed** en mettant à jour l'état de fin d'exécution de l'opération ($OpComp := true$) et de la disponibilité du job de celle-ci.

Le patron de ressource (fig.2b) modélise le comportement d'une ressource r . Dans la localité **Idle** initiale, le modèle *Ressource* r attend une requête d'exécution ($Req(r)$). Lors du franchissement de la transition, l'horloge locale est initialisée à zéro ($x_r := 0$) et l'identifiant de l'opération est enregistrée dans la variable o_{jk} . Dans la localité **Executing Op**, la ressource lance l'exécution de l'opération. L'invariant ($x_r \leq d_{jkr}$) assure que la durée ne dépassera pas d_{jkr} et lorsque la valeur de la durée est atteinte, la garde [$x_r == d_{jkr}$] est alors satisfaite et l'événement de synchronisation $Comp(o_{jk})$ se déclenche permettant à la ressource de retourner à sa localité initiale **Idle**.



(a) Patron d'une opération



(b) Patron d'une ressource

Figure 2: Patrons de modélisation d'un ordonnancement

Les patrons présentés (fig.2) partagent les informations en utilisant des événements de synchronisation (*Req*, *Comp*) urgents, et des variables partagées (*operation*). La prochaine partie détaille la modélisation des perturbations et leur intégration avec les patrons d'ordonnement.

4.2 Modélisation des perturbations

Nous faisons l'hypothèse que les perturbations considérées sont indépendantes. Toute perturbation génère une variation sur la durée d'exécution des opérations et donc sur le Makespan. La durée d'exécution d'une opération affectée par des perturbations, est exprimée par (2). Chaque durée d'exécution de l'opération d_{jkr} est définie par une durée d'exécution de référence d_{jkr}^{ref} donnée par les données déterministes et une somme de fluctuations liées à l'ensemble de perturbations (*Pert*) considérées comme des incertitudes (*U*) ou des aléas (*H*).

De plus on distinguera les incertitudes liées à la durée d'exécution (U^{ex}) et les incertitudes liées à l'occurrence des aléas (U^H). Les incertitudes liées aux durées d'exécution sont la somme des différentes fluctuations générées par chaque incertitude u appartenant à l'ensemble U^{ex} ($\sum_{u=1}^{NbU^{ex}} \delta d_{jkr}^u$). Pour les aléas, nous faisons l'hypothèse que chaque opération peut être affectée par un ou plusieurs aléas. Chaque aléa $h \in H$ est lié à: une durée dH_{jkr}^h qui reflète le retard généré et une probabilité $p(h, o_{jk})$ qui représente la probabilité que l'aléa h impacte l'opération o_{jk} . À partir de cette probabilité, une variable booléenne H_{jk}^h peut être construite avec $H_{jk}^h = 1$ si l'opération o_{jk} est effectivement impactée par l'aléa h . Les retards dus aux aléas de H peuvent donc être écrits comme $\sum_{h=1}^{NbH} H_{jk}^h dH_{jkr}^h$.

Le fait que chaque aléa soit lié à une durée permet de prendre en compte également des incertitudes sur cette durée représentées par l'ensemble U^h (avec $U^H = \bigcup_{h \in H} U^h$). En effet, nous considérons que la durée dH_{jkr}^h générée par l'aléa est d'abord une durée de référence connue $dH_{jkr}^{ref,h}$ et une somme de fluctuations $\sum_{u=1}^{NbU^h} \delta d_{jkr}^u$ dues aux incertitudes sur cette durée comme pour les incertitudes d'exécution. Au final, la durée d'exécution d_{jkr} permettant de prendre en

compte les perturbations peut s'écrire comme dans l'équation (2).

$$d_{jkr} = d_{jkr}^{ref} + \sum_{u=1}^{NbU^{ex}} \delta d_{jkr}^{U^{ex}} + \sum_{h=1}^{NbH} H_{jk}^h \times \left(dH_{jkr}^{ref,h} + \sum_{u=1}^{NbU^h} \delta d_{jkr}^{U^h} \right) \quad (2)$$

Les variables de l'expression (2) sont des variables aléatoires qui suivent des distributions de probabilité connues (seule d_{jkr}^{ref} est considérée comme déterministe et statique). Toutes les fluctuations exprimées par la variable δd sont considérées comme appartenant à des intervalles $[\delta d^-, \delta d^+]$. Suite à la description donnée des perturbations (2), nous présentons dans les sous-sections suivantes une approche de modélisation adaptée.

4.2.1 Patron de modélisation d'une incertitude

Les patrons présentés dans (fig. 3a) proposent une modélisation du comportement des perturbations qui vient s'insérer dans le modèle d'initialisation (fig. 4).

Le patron présenté (fig.3a) permet de modéliser le comportement des incertitudes que ce soit U^{ex} ou U^H . Il permet la représentation de la distribution de probabilité discrétisée suivie. Suivant les notations données dans l'expression 2, on note δd le paramètre de la durée associé à une variable aléatoire discrète X_d . Le pas de discrétisation de la distribution de probabilité est t et l représente le compteur d'itération exécutée. L'idée de ce modèle est d'augmenter itérativement, la valeur de δd dans l'intervalle $[\delta d^-, \delta d^+]$.

Depuis la première localité de (fig.3a), δd est initialisé à δd^- et l est initialisée à zéro ($l := 0$). En arrivant à la localité **Prob**, deux transitions sont possibles en fonction de la satisfaction de leurs gardes. La première garde modélise le cas où δd est inférieur à δd^+ ($[\delta d < \delta d^+]$). Lorsque cette garde est vraie, deux évolutions probabilistes sont possibles. Avec la probabilité $p(l)$ la localité finale **EndProb** est atteinte et δd garde sa dernière la valeur. Sinon, avec la probabilité $1 - p(l)$, l'automate retourne à la localité **Prob** avec mise à jour du paramètre d'itération l à $l + 1$ ainsi que la mise à jour en $\delta d = \delta d + t$ de sorte à augmenter la valeur de δd d'un pas t de discrétisation. De là, un autre pas de discrétisation est exécutée. La probabilité que $X_d = \delta d$ est la probabilité de boucler dans le patron $l - 1$ fois et de sortir de la boucle lors de la l -ième itération.

Si la fluctuation maximale est atteinte ($[\delta d == \delta d^+]$), l'automate évolue directement vers la localité **EndProb** et la valeur du δd devient δd^+ .

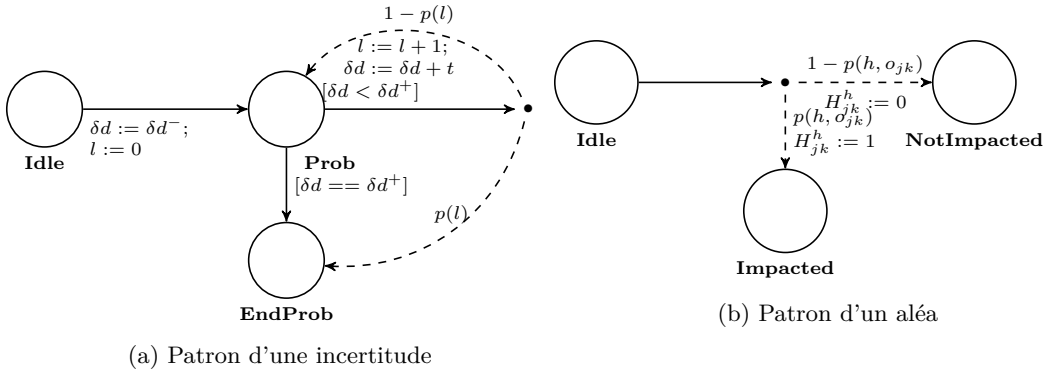


Figure 3: Patrons stochastiques de perturbations

Les caractéristiques recueillies auprès du décideur permettent de définir les perturbations comme des variables aléatoires. Une variable aléatoire est liée à une distribution de probabilité connue ayant une fonction de répartition définie ($F_X(x)$). Nous pouvons démontrer que $p(l)$ peut être calculée en fonction de $F_X(x)$ comme dans la proposition (1).

Proposition 1. *la valeur $p(l)$ est exprimée en fonction d'une distribution de probabilité tronquée dans l'intervalle $[x^{min}, x^{max} + 1[$.*

$$\begin{cases} p(0) = \frac{F_X(x^{max}+1) - F_X(x^{min})}{F_X(x^{max}) - F_X(x^{min})} & \text{for } l = 0 \\ p(l) = \frac{F_X(x^{min}+l+1) - F_X(x^{min}+l)}{F_X(x^{min}+l) - F_X(x^{min}+l-1)} \cdot \frac{p(l-1)}{1-p(l-1)} & \text{for } l \geq 1 \end{cases} \quad (3)$$

4.2.2 Patron de modélisation d'un aléa

Le patron (fig. 3b) représente le comportement d'un aléa. Pour rappel, chaque opération o_{jk} peut être impactée par un aléa h de H avec une probabilité connue $p(h, o_{jk})$. Depuis la localité **Idle**, (i) soit l'opération est affectée avec la probabilité $p(h, o_{jk})$ et le patron passe à la localité **Impacted** après que la variable d'aléa (H_{jk}^h) soit mise à 1; (ii) soit l'opération n'est pas impactée par l'aléa avec la probabilité $1 - p(h, o_{jk})$ et le patron passe dans la localité **NotImpacted** après que la variable H_{jk}^h ait été mise à 0.

4.3 Initialisation des perturbations

Le modèle d'initialisation permet l'intégration des différentes perturbations (fig.3a) et (fig.3b) dans l'approche de modélisation sans affecter les autres modèles. A partir de la localité initiale **Begin Init**, deux évolutions sont possibles. Si la perturbation est considérée, la valeur du booléen $Pert_i$ est à 1 ($[Pert_i == 1]$). Le patron de perturbation est ensuite associé au modèle, qu'il s'agisse d'une incertitude (fig.3a) ou d'un aléa (fig.3b). Lorsque le patron de perturbation termine son exécution, le modèle évolue vers la localité **End Init** en mettant à jour le nombre de perturbations considérées au moyen du compteur c ($c := c + 1$). Si la perturbation n'est pas prise en compte, la valeur de $Pert_i$ est à 0 ($[Pert_i == 0]$) et le modèle passe directement à la localité **End Init**.

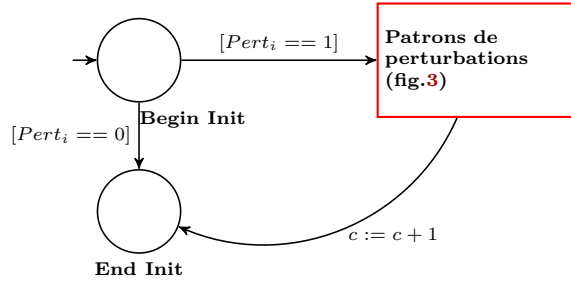


Figure 4: Modèle d'initialisation des perturbations

A partir des données fournies par le décideur au processus d'évaluation (fig.1), une procédure de prise en compte des perturbations est définie :

- La première étape permet de définir les perturbations considérées $Pert$ ($Pert = \{U, H\}$). Le nombre de chaque type de perturbations NbU^{ex} , NbH et $NbU^H = \sum_{h \in H} NbU^h$ peut être calculé.
- La deuxième étape consiste à instancier les modèles de perturbations. Dans le cas d'incertitudes, le paramètre δd et ses bornes δd^- et δd^+ sont instanciés selon le type de durée impactée par l'incertitude (durée d'exécution, durée de réparation, etc.). La probabilité du modèle $p(l)$ est également calculée en fonction des données probabilistes fournies par le décideur et des données de l'atelier. Dans le cas des aléas, la probabilité $p(h, o_{jk})$ est également déterminée en fonction des données d'entrée.
- La dernière étape consiste à instancier les modèles en fonction des données de l'ordonnement (nombre d'opérations et de ressources) ainsi que le nombre de perturbations considérées.

Le nombre de modèles de perturbation à instancier, dépend du nombre d'opérations $NbOp$, et du nombre de perturbations. Le nombre de modèles (NbM) nécessaires pour traiter le problème de perturbations peut être déduit de (2) et est exprimé par (4).

$$NbM = NbOp \times (NbU^{ex} + NbH + NbU^H) \quad (4)$$

4.4 Intégration des perturbations à l'ordonnement

L'initialisation des modèles de perturbations permet l'initiation des différentes fluctuations générées par les perturbations. Le lien entre les modèles de perturbations et ceux d'ordonnement se fait en ajoutant une transition au modèle d'opération (fig.2a). En effet, la localité initiale devient la localité nommée **Waiting** où le modèle *opération* attend que toutes les perturbations soient initialisées, c'est-à-dire que la garde $[c == NbM]$ soit vérifiée. Ensuite, la durée est mise à jour en tenant compte de l'expression (2). Après avoir franchi la transition, le patron d'opération est exécuté comme expliqué dans la figure (2a).

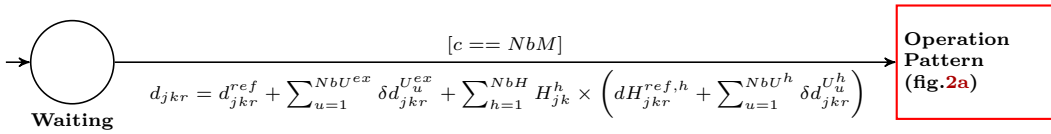


Figure 5: Considération des perturbations dans le modèle d'opération

Cette section décrit la phase de modélisation du processus d'évaluation. La phase suivante du processus d'évaluation consiste à vérifier la propriété qui permettra de mesurer les niveaux de robustesse.

5 Étape d'évaluation de la robustesse

L'évaluation du niveau de robustesse défini dans l'expression (1) est vérifié en utilisant un model-checker. Dans le domaine SED, un model-checker est généralement utilisée pour vérifier si un modèle satisfait ou non certaines propriétés. Ces propriétés peuvent être exprimées dans

plusieurs langages logiques (CTL, PCTL, etc.). Pour évaluer la robustesse, nous utilisons une extension du model-checking pour les modèles STA qui sont exprimées en logique PCTL (Probabilistic Computation Tree Logic). Ce langage est une extension probabiliste de CTL (Computation Tree Logic) [3]. Ce type de logique permet d'exprimer des propriétés telle que "Quelle est la probabilité que le modèle soit à l'état A , dans l'intervalle précis $[0, T]$?". Cette question peut être transcrite en PCTL comme dans l'expression: $P = ?[F \leq T \text{ "A"}]$. L'opérateur $P = ?$ exprime l'évaluation de probabilité Pr , l'opérateur F signifie qu'il existe éventuellement un chemin où l'automate est à l'état "A".

Pour évaluer le niveau de robustesse d'un ordonnancement, la question posée ci-dessus devient: "Quelle est la probabilité que tous les chemins mènent à un état global où tous les modèles d'opérations sont dans leur localité marquée **Completed** avec une durée qui est inférieure ou égale à une deadline donnée \tilde{d} ". En utilisant la logique PCTL, la formule (1) peut être exprimée comme la propriété (5) où le Makespan $C_{max}(S_i, Pert)$ est donné par la valeur de l'horloge globale (clk).

$$P = ?[F \leq \tilde{d} \text{ "All operations } o_{jk} \text{ are Completed"}] \quad (5)$$

Plusieurs méthodes et outils existent pour vérifier les modèles stochastiques. Deux familles peuvent être distinguées : le Model Checking numérique (NMC) et le Model Checking statistique (SMC). D'une part, les NMC utilisent des méthodes d'évaluation exactes afin de déterminer la valeur de probabilité d'une propriété. Ce type de vérification garantit la précision de la solution donnée, mais il n'est pas adapté aux problèmes de grande taille. En réalité, lorsque la taille du système à vérifier est relativement grande, le NMC fait face à un problème d'explosion combinatoire.

D'autre part, le SMC génère différents chemins d'exécution et vérifie, après chaque exécution, la satisfaction d'une propriété ainsi que les résultats statistiques associés (comme pour la simulation Monte Carlo). Ceci évite l'explosion combinatoire [4]. Dans cette optique, nous avons choisi de l'utiliser pour évaluer les niveaux de robustesse des ordonnancements. L'approche par solution SMC utilise la méthode de l'intervalle de confiance afin de se rapprocher de la valeur de la probabilité. Un intervalle de confiance (IC) est un intervalle estimé ayant une précision spécifiée ϵ et un niveau de confiance α tel que, la probabilité RL_i appartient à l'intervalle $[RL_i - \epsilon; RL_i + \epsilon]$ avec une chance de $100 \times (1 - \alpha)\%$ [19].

Dans la section suivante, nous combinons les modèles STA présentés dans la section précédente avec la méthode SMC pour évaluer les niveaux de robustesse dans un intervalle précis avec un niveau de confiance de $\alpha = 2$ et une précision $\epsilon = 0.01$. Le Model-Checker SMC évaluera la formule PCTL (5) en exécutant les N itérations.

6 Etude de cas

L'approche présentée dans cet article est expérimentée sur une étude de cas académique présentée dans [8]. Le but de l'expérimentation est de démontrer la faisabilité de l'approche et de mettre en œuvre le processus d'évaluation (fig. 1). Le système de production étudié est un Job-Shop Flexible avec 8 jobs et 7 ressources. L'atelier comprend un tour CNC (M1), une rectifieuse (M2), trois fraiseuses (M3 à M5) et deux fours (M6 et M7). Le décideur a le choix entre trois ordonnancements $S_i = \{S_1, S_2, S_3\}$. Les données déterministes sont extraites des diagrammes de Gantt (fig.6) : i) Makespans déterministes (C_{max}^{ref}) et ii) les durées de référence (durées d'exécution (d_{jkr}^{ref}) et de réparation ($d_{rep}^{ref} = 7$)). Les perturbations prises en compte par le décideur sont : i) une incertitude sur les durées d'exécution et de réparation

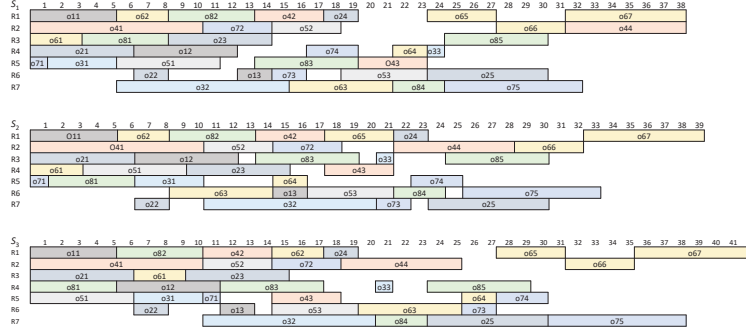


Figure 6: Ordonnements déterministes à évaluer

($\delta d^- = -25\%d^{ref}$ et $\delta d^+ = +25\%d^{ref}$). ii) un aléas de pannes machines. Le décideur fixe la deadline souhaitée à $\tilde{d} = 110\%C_{max}^{ref}(S_i)$.

6.1 Modélisation de l'étude de cas

Les perturbations qui sont habituellement traitées, dans la littérature, sont des incertitudes sur les durées (notamment sur les durées d'exécution des opérations) et les pannes ressources. Pour démontrer l'applicabilité de notre approche et de notre procédure, nous considérons que l'ensemble de perturbations est $Pert = \{u_1, u_2, h\}$, avec u_1 l'incertitude sur la durée d'exécution des opérations, u_2 l'incertitude sur la durée des réparations et h l'occurrence de pannes sur les ressources. L'application de l'expression (2) définit la durée d_{jkr} dans (6).

$$d_{jkr} = d_{jkr}^{ref} + \delta d_{jkr}^{u_1} + H^{h,jk} [dH_{jkr}^{ref,h} + \delta d_{jkr}^{u_2^h}] \quad (6)$$

Les modèles de perturbations (fig.3a) sont instanciés en fonction des paramètres des perturbations. Dans un premier temps, pour l'incertitude sur la durée d'exécution u_1 , on considère que la fluctuation est notée δd_{jkr} . Cette fluctuation suit une loi de probabilité exponentielle tronquée dans l'intervalle donné $[\delta d_{jkr}^-, \delta d_{jkr}^+]$ et ainsi les bornes de la durée d'exécution qui définies par $d_{jkr}^{min} = d_{jkr}^{ref} - \delta d_{jkr}^-$ and $d_{jkr}^{max} = d_{jkr}^{ref} + \delta d_{jkr}^+$. Dans un deuxième temps, l'incertitude sur la durée de réparation u_2 est traitée de la même manière que celle des durées d'exécution. Nous notons la fluctuation de la durée de la réparation δd_{rep} donnée dans un intervalle connu $[\delta d_{rep}^-, \delta d_{rep}^+]$. Pour les deux incertitudes, les valeurs de $p(l)$ sont obtenues en considérant la fonction de répartition de la distribution exponentielle dans (3). Le modèle des aléas (fig.3b) est instancié pour h . Chaque opération a une probabilité $p(h, o_{jk})$ d'être affectée par la défaillance de la ressource. Cette probabilité est connue et donnée par le décideur. Dans notre cas, $p(h, o_{jk})$ est égale à 3%. Cette valeur est choisie pour exprimer le fait, qu'en moyenne, une opération est affectée par une défaillance dans l'horizon de l'ordonnement. Dans ce cas, le nombre de modèles nécessaires pour traiter toutes les perturbations est $NbM = 35 \times (1 + 1 + 1 \times 1) = 105$ modèles, sachant que $NbU^{ex} = 1$, $NbH = 1$ et $NbU^H = 1$.

6.2 Évaluation de la robustesse

La deuxième étape du processus d'évaluation est l'évaluation de la robustesse des ordonnancements, à partir de la section (5) on obtient la formule PCTL qui permet d'évaluer la ro-

bustesse d'un ordonnancement (expression. 5). L'implémentation de cette formule dans le Model-Checker UppAal SMC (7) permet l'évaluation des modèles STA implémentés.

$$Pr[\leq \tilde{d}](<> forall(i : int[1, NbOp])Operation(i).Completed) \quad (7)$$

La formule permet d'évaluer la probabilité que toutes les opérations soient dans leur localité marquée **Completed** dans une durée totale d'exécution inférieure ou égale à la deadline $\tilde{d} = 110\%C_{max}^{ref}(S_i)$ (respectivement 42, 43 et 46 pour S_1 , S_2 et S_3).

6.3 Implémentation des scénarios

Dans cet article, nous étudions cinq scénarios avec différentes perturbations. Chaque scénario traite une ou plusieurs perturbations. Pour chaque scénario, les seuls changements dans les modèles sont les valeurs des booléens $Pert_i$ qui décident si la perturbation est considérée ($Pert_i = 1$) ou non ($Pert_i = 0$).

- Scénario 1 : incertitude sur la durée d'exécution ($u_1 = 1, h = 0, u_2 = 0$),
- Scénario 2 : aléa dû à la panne de ressources ($u_1 = 0, h = 1, u_2 = 0$),
- Scénario 3 : aléa dû à la panne de ressources et incertitudes sur la durée des réparations ($u_1 = 0, h = 1, u_2 = 1$),
- Scénario 4 : incertitudes sur la durée d'exécution des opérations et aléa lié à la panne de ressources ($u_1 = 1, h = 1, u_2 = 0$).
- Scénario 5 : Incertitudes sur la durée d'exécution, avec aléa dû à la panne de ressources et incertitudes sur la durée de la réparation ($u_1 = 1, h = 1, u_2 = 1$).

Les modèles et propriétés présentés dans les sections ci-dessus (4, 5) sont implémentés dans l'outil UppAal SMC. Les paramètres du Model-Checker de modèle sont un niveau de confiance de $\alpha = 2$ et une précision $\epsilon = 0.01$.

6.4 Expérimentation et discussion

Pour les cinq scénarios, nous rassemblons les différents niveaux de robustesse obtenus dans le tableau 1. Grâce à ces informations, le décideur peut interpréter les résultats et choisir un ordonnancement à exécuter dans l'atelier. A titre d'exemple, pour le troisième scénario, l'ordonnancement S_2 peut être choisi car il présente le niveau de robustesse le plus grand. Mais il a surtout l'avantage de présenter un bon compromis robustesse/Makespan. Le même constat peut être fait sur l'ensemble des scénarios.

Dans cette application, le décideur avait trois ordonnancements déterministes avec des makespans déterministes proches $C_{max}^{ref}(S_i)$ mais des séquences différentes. Notre approche a permis d'évaluer les différentes perturbations qui peuvent se produire dans ces ordonnancements et de déterminer celui qui les absorbe le mieux. Cela donne également des informations sur la résistance des ordonnancements à une combinaison de perturbations. Ce type d'information est important dans le processus de prise de décision, elle permet une prise en compte préventive d'éventuelles perturbations. L'adaptabilité de l'approche est démontrée par les différents scénarios étudiés à partir d'un seul et même problème, ce qui donne au décideur la liberté de combiner et d'évaluer différentes perturbations avec des deadline différentes. En ajoutant d'autres perturbations qui ont un impact sur la durée d'exécution des opérations, il n'est pas nécessaire de faire un effort important de modélisation.

Table 1: Résultats d'évaluation de la robustesse

| Scénarios | Ordonnancement | RL_i |
|------------|----------------|--------------|
| Scénario 1 | S_1 | [0.94, 0.96] |
| | S_2 | [0.97, 0.99] |
| | S_3 | [0.96, 0.98] |
| Scénario 2 | S_1 | [0.58, 0.60] |
| | S_2 | [0.64, 0.66] |
| | S_3 | [0.61, 0.63] |
| Scénario 3 | S_1 | [0.70, 0.72] |
| | S_2 | [0.73, 0.75] |
| | S_3 | [0.71, 0.73] |
| Scénario 4 | S_1 | [0.65, 0.67] |
| | S_2 | [0.71, 0.73] |
| | S_3 | [0.68, 0.70] |
| Scénario 5 | S_1 | [0.69, 0.71] |
| | S_2 | [0.75, 0.77] |
| | S_3 | [0.71, 0.73] |

7 Conclusions et perspectives

Dans cet article, nous proposons une approche générique pour évaluer la robustesse d'ordonnancements soumis à des perturbations. Nous avons répondu à deux questions de recherche importantes : comment modéliser le problème de l'ordonnancement sous perturbations et comment mesurer la robustesse des ordonnancements devant être évalués. Cette approche peut être utilisée pour faire face à des perturbations qui ont un impact sur la durée d'exécution des opérations et choisir l'ordonnancement qui est robuste devant une ou plusieurs des perturbations définies. Les sections 3 et 4 proposent une approche basée sur les automates temporisés stochastiques pour la modélisation des ordonnancements et perturbations. La généralité et l'adaptabilité de l'approche sont ses principaux atouts. En effet, le cadre de modélisation proposé est générique face aux types d'ateliers (Job Shop, Flow Shop, Open Shop, etc.). Il permet également une adaptabilité aisée lors du traitement de nouvelles perturbations. Ce qui est une caractéristique importante face aux perturbations dans le contexte de l'Industrie 4.0 et aussi dans le contexte de la prise de décision. La section 5 propose une approche pour aider le décideur à évaluer différents ordonnancements devant différentes perturbations. Cette approche utilise le Model Checking statistique pour évaluer formellement les niveaux de robustesse des ordonnancements. L'application présentée dans cet article dans la section (6), a d'abord démontré la faisabilité de l'approche proposée. Elle montre également l'importance du processus d'évaluation dans la prise de décision au sein d'un atelier et permet une analyse différente du comportement d'un ordonnancement devant les perturbations qui émanent du système et de la production. Les perturbations traitées dans cet article sont les incertitudes sur les durées d'exploitation et de réparation et les aléas liés à la pannes de ressources, qui sont typiquement des perturbations liées au système de production. Dans les futurs travaux, l'approche devra être étendue à l'autre type de perturbations générées par l'environnement de production (demande client, rupture d'approvisionnement, etc.). En effet, l'une des caractéristiques de la production dans l'industrie 4.0 à considérer est la personnalisation des produits et la flexibilité en fonction de la demande du client. De plus, le passage à l'échelle de notre approche devra être testé pour des systèmes de production de taille réelle et l'amélioration de l'accessibilité du processus d'évaluation à intégrer dans un processus d'aide à la décision devra être évaluée avec un panel d'utilisateurs.

References

- [1] Yasmina Abdeddaïm, Eugene Asarin, and Oded Maler. Scheduling with timed automata. *Theoretical Computer Science*, 354(2):272–300, 2006.
- [2] Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.
- [3] C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.
- [4] P. Ballarini, H. Djafri, M. DufLOT, S. Haddad, and N. Pekergin. Petri nets compositional modelling and verification of flexible manufacturing systems. *Proceeding of the 7th Conference on Automation Science and Engineering*, 2011.
- [5] Jean-Charles Billaut, Aziz Moukrim, and Eric Sanlaville. *Flexibility and robustness in scheduling*. John Wiley & Sons, 2013.
- [6] Stéphane Dauzères-Pérès, Philippe Castagliola, and Chams Lahlou. Service Level in Scheduling. *Flexibility and Robustness in Scheduling*, pages 99–121, 2010.
- [7] Stephanie Geist, Dmitry Gromov, and Jörg Raisch. Timed discrete event control of parallel production lines with continuous outputs. *Discrete Event Dynamic Systems*, 18(2):241–262, 2008.
- [8] Vincent Giard. *Gestion de la production et des flux*. Economica, 2003.
- [9] Davide Giglio, Riccardo Minciardi, Simona Sacone, and Silvia Siri. Optimal control of production processes with variable execution times. *Discrete Event Dynamic Systems*, 19(3):423–448, 2009.
- [10] Sara Himmiche, Pascale Marangé, Alexis Aubry, Marie DufLOT, and Jean-François Pétin. Evaluation de la robustesse d'un ordonnancement par Automates Temporisés Stochastiques. 11ème Colloque sur la Modélisation des Systèmes Réactifs, MSR 2017, November 2017.
- [11] Sara Himmiche, Pascale Marangé, Alexis Aubry, and Jean-François Pétin. Robust production scheduling under machine failures-a des based evaluation approach. *IFAC-PapersOnLine*, 51(7):271–276, 2018.
- [12] Avenir Kobetski and Martin Fabian. Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming. *Discrete Event Dynamic Systems*, 19(3):287–315, Sep 2009.
- [13] Panos Kouvelis and Gang Yu. *Robut Discrete Optimization and its Applications*. Springer, 1997.
- [14] Kim G Larsen, Paul Pettersson, and Wang Yi. Uppaal in a nutshell. *International journal on software tools for technology transfer*, 1(1-2):134–152, 1997.
- [15] Haralambie Leahu, Bernd Heidergott, and Arie Hordijk. Perturbation analysis of waiting times in the g/g/1 queue. *Discrete Event Dynamic Systems*, 23(3):277–305, 2013.
- [16] Dimitri Lefebvre and Gonzalo Mejia. Robust scheduling in uncertain environment with Petri nets and beam search. *IFAC-PapersOnLine*, 51(11):1077–1082, 2018.
- [17] Zukui Li and Marianthi Ierapetritou. Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, 32(4):715 – 727, 2008.
- [18] Pascale Marangé, Alexis Aubry, and Jean-François Pétin. Ordonnancement d'ateliers à partir de patrons de modélisation basés sur des automates communicants. 10ème Colloque sur la Modélisation des Systèmes Réactifs, MSR 2015, November 2015. Poster.
- [19] Vincent Nimal. *Statistical approaches for probabilistic model checking*. PhD thesis, University of Oxford, 2010.
- [20] Sebastian Panek, Sebastian Engell, and Olaf Stursberg. Scheduling and planning with timed automata. 21:1973 – 1978, 2006.
- [21] Sebastian Panek, Olaf Stursberg, and Sebastian Engell. Efficient synthesis of production schedules by optimization of timed automata. *Control Engineering Practice*, 14(10):1183–1197, 2006.