Évaluation de la robustesse d'un ordonnancement par Automates Temporisés Stochastiques

Sara Himmiche[1,2], Pascale Marangé[1,2], Alexis Aubry[1,2], Marie Duflot[3], 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
[3] Université de Lorraine, CNRS et Inria, LORIA UMR 7503, F-54000 Nancy
marie.duflot-kremer@loria.fr

Abstract

Les modèles et outils des Systèmes à Evénéments Discrets (SED) ont montré leur apport et leur efficacité pour la modélisation et la résolution de problèmes d'ordonnancement dans le domaine des systèmes manufacturiers de production. Leur principal atout réside dans leur capacité à appréhender naturellement les dynamiques sous-jacentes aux ressources de production ainsi que les logiques de configuration des ateliers (Job-shop, Flow-shop, Open-shop, hybrides...).

De plus, les extensions stochastiques des modèles de SED offrent d'intéressantes perspectives pour la prise en compte de l'incertain en ordonnancement : incertitudes sur les ressources (durée opératoires, aléas de fonctionnement, pannes...) mais aussi sur la demande (variabilité importante des produits, personnalisation de masse...). L'objectif de cet article est de démontrer la faisabilité d'une approche basée sur les automates temporisés stochastiques et sur des techniques de model-checking statistique pour évaluer la robustesse d'un ordonnancement face à des aléas en se restreignant, dans le cadre de cette étude, aux incertitudes sur les durées opératoires.

1 Introduction

L'ordonnancement de tâches consiste à définir les dates de début et d'achèvement d'une séquence donnée de tâches, et à affecter des ressources dédiées pour produire un service donné. Dans le contexte des systèmes manufacturiers de production, les tâches peuvent être des opérations de fabrication (ou de maintenance), effectuées sur des machines (en utilisant des outils spécifiques), pour obtenir un ensemble de produits. Les approches permettant d'obtenir un ordonnancement, que l'on cherche généralement à rendre optimal en termes de durée de fabrication de l'ensemble de produits ou en termes d'utilisation des ressources, font l'objet d'une littérature scientifique abondante, notamment issue de la recherche opérationnelle. Ces dernières années, les méthodes et les langages basés sur la théorie des SED ont apporté une alternative réaliste pour l'ordonnancement de la production [11][16].

Ces différentes approches sont généralement réalisées de manière prévisionnelle en considérant un environnement stable (que ce soit au niveau de la demande et des ressources). L'ordonnancement optimal, calculé selon ces hypothèses optimistes, risque de voir ses performances détériorées lors de sa mise en œuvre, compte tenu des écarts inévitables en pratique vis à vis de l'environnement de référence : incertitudes et forte variabilité sur les ensembles de produits à réaliser (volume et mix, personnalisation de masse, production de petites séries, délais de plus en plus courts entre prise de commande et délai de livraison...), incertitudes sur les ressources de production (durées opératoires, pannes des machines, aléas de fabrication...).

Plusieurs réponses peuvent être proposées pour répondre à ce problème. La première consiste à mettre en œuvre un ordonnancement réactif, calculé en ligne pendant la production pour tenir compte de la réalité du terrain quitte à réduire les exigences en termes d'optimalité. Généralement les approches actuelles reposent sur un ordonnancement prévisionnel que l'on adapte (ou que l'on dégrade) en le couplant à un ordonnancement réactif. Une deuxième réponse consiste à adopter une approche proactive, en déterminant de manière prévisionnelle un ensemble d'ordonnancements admissibles (non nécessairement optimaux) dont on cherche à évaluer la robustesse face à des aléas. Cela suppose que les perturbations attendues soient introduites dans le modèle : les variables incertaines du problème suivent des lois de probabilité (modèle stochastique) ou appartiennent à des intervalles (continus, discrets ou flous). On privilégie, dans cette approche, des performances globales (notion de robustesse) sur un ensemble de données (loi de probabilité, intervalle) plutôt que la seule performance locale (notion d'optimalité) sur des données statiques fixées (une seule valeur à chaque variable du problème). La contribution présentée dans cet article entre dans le cadre de cette famille d'approches proactives.

L'objectif de cet article est de présenter une approche permettant d'évaluer la robustesse d'un ordonnancement face à des incertitudes sur le temps d'exécution d'une opération sur une machine et de déterminer la durée minimale d'un ordonnancement requise pour satisfaire une exigence donnée en termes de niveau de robustesse. Nous cherchons à vérifier qu'il est faisable d'utiliser les SED pour résoudre des problèmes de robustesse en ordonnancement.

La suite de l'article est organisée en trois parties. La section 2 présente le problème d'ordonnancement en contexte incertain et définit les principaux indicateurs permettant de caractériser la robustesse d'un ordonnancement soumis à des perturbations. Dans la troisième section, une approche basée sur les automates stochastiques et des techniques de model-checking statistique est proposée. La section 4 illustre la proposition sur un exemple académique. Enfin, cet article se conclut par un bilan de l'approche proposée et des perspectives ouvertes.

2 Définition du problème

2.1 Ordonnancement d'ateliers flexibles

Classiquement, l'ordonnancement de la production consiste à (i) affecter les opérations à réaliser à des ressources et (ii) à définir le séquencement de ces opérations sur les ressources (c'est-à-dire définir l'ordre des opérations sur une machine), en respectant des contraintes inhérentes au système de production considéré et en optimisant un critère (durée totale, nombre d'opérations en retard ...). De plus, lorsque les durées d'exécution opératoire sont considérées comme parfaitement connues et statiques, il devient possible de fixer des dates de début et de fin à chaque opération.

Le critère d'optimisation le plus souvent traité est la durée totale de l'ordonnancement (le makespan, noté C_{max}).

Formellement, un problème d'ordonnancement consiste à trouver un ordonnancement $\mathcal{S}$ qui optimise un critère d'optimalité z tout en vérifiant des contraintes (contraintes de précédence entre opérations, ...). Dans le cas d'un problème déterministe, tous les paramètres sont connus et statiques.

Dans le contexte particulier des ateliers de production, on considère un ensemble de produits J devant être fabriqués sur un ensemble de machines M (le nombre de machines est donné par card(M) et est noté $\mathcal{M}$) suivant une gamme de fabrication O^J. L'exécution de l'opération o_{jk} (k^{ieme} opération de la gamme O_j^J du produit j) requiert une machine m qualifiée pour cette

opération o_{jk} exécutée pendant une durée d_{jkm}. Le nombre d'opérations à exécuter dans l'atelier est donné par $\sum_J |O^J|$ et est noté $\mathcal{O}$) Il existe différents types d'ateliers classiques caractérisant les flux possibles de produits : Flow-shop, Job-shop, Open-shop. Nous nous intéressons aux ateliers flexibles qui sont caractérisés par le fait qu'une opération peut être réalisée sur plusieurs machines et que les machines peuvent réaliser plusieurs opérations. Dans le cadre d'un atelier Flow-shop flexible, un ensemble d'étapes de fabrication est défini. Chaque étape est supportée par un ensemble de machines pouvant réaliser un ensemble d'opérations et doit être suivie par tous les produits. Pour les ateliers de type Job-shop, chaque produit possède sa propre gamme de fabrication (ils ne suivent donc pas forcément le même chemin dans l'atelier). Dans le cadre des ateliers de type Open-shop, les opérations d'une gamme peuvent être exécutées dans un ordre quelconque.

La différence de type d'atelier se traduit par des contraintes de précédence différentes à respecter pour les opérations :

- Cas des ateliers Job-shop et Flow-shop : l'ordre d'exécution des opérations d'un produit j est fixé par sa gamme, c'est-à-dire que pour tout k l'opération o_{jk} doit être exécutée avant l'opération o_{jk+1}
- Cas d'un atelier Open-shop : l'ordre d'exécution des opérations d'un produit j n'est pas défini, mais les opérations ne peuvent pas s'exécuter au même instant, c'est à dire que pour tout $k \neq l$ alors l'opération o_{jk} doit être finie avant que l'opération o_{jl} ne commence ou inversement.
- Des cas hybrides mixant les 3 types d'atelier existent.

2.2 Prise en compte des perturbations en ordonnancement

2.2.1 Définition des perturbations

Le problème d'ordonnancement des ateliers de production considère souvent que ces derniers évoluent dans un environnement stable et certain. Cette hypothèse n'est généralement plus raisonnable au vu de l'évolution des ateliers de production et de leur environnement. En effet, un atelier est soumis à plusieurs types de perturbations qui peuvent être internes (issues du processus de production lui même) ou externes (issues de l'environnement du processus). Les perturbations peuvent êtres séparées en deux catégories:

- Les aléas : ils sont définis comme étant l'occurrence d'événements imprévus dans le processus de production ou son environnement (panne machine, rupture de stock...).
- Les incertitudes : elles sont définies comme étant la différence entre la quantité d'information requise pour exécuter une tâche et la quantité d'information présente (les incertitudes sur les durées d'exécution des opérations, les incertitudes sur la demande...)

Dans [5], différents modèles permettant de modéliser des perturbations sont présentés.2017-09-29 Le modèle stochastique, par exemple, considère que certaines données d'entrée du problème (durées opératoires, date d'arrivée d'un événement) peuvent être modélisées par des variables aléatoires. Ce modèle nécessite donc de disposer d'un historique de données suffisant afin d'avoir un modèle pertinent.

Un autre modèle consiste à dire que la donnée incertaine peut être considérée dans un ensemble discret ou continu de scénarios. Chaque scénario fixe une valeur aux données incertaines. Il est possible d'associer une probabilité à chaque scénario (même si elle n'est pas calculable directement) : on retrouve le modèle stochastique.

Dans les travaux de cet article, la perturbation considérée est l'incertitude sur la durée d'exécution d'une opération o_{jk} sur une machine m. Cette durée peut varier dans un intervalle connu. Une durée de référence d_{jkm}^{ref} peut souvent être donnée par les chefs d'atelier. Pour considérer d'éventuelles incertitudes sur la durée d'exécution d_{jkm}, il est nécessaire de considérer un intervalle contenant la valeur de référence.

$$d_{jkm} \in [d_{jkm}^{min}, d_{jkm}^{max}] \tag{1}$$

Nous ferons l'hypothèse que la durée réelle d_{jkm} est une variable stochastique décrite par une distribution de probabilité sur l'intervalle, et I représente le vecteur aléatoire des durées.

$$I = (d_{jkm})_{j\in J,\ k\in O,\ m\in M} \tag{2}$$

Dans cette étude, nous supposons que les gammes des produits sont parfaitement connues et ne varient pas et que les machines ne subissent pas d'aléas et sont toujours disponibles. La prise en compte de la perturbation sur la durée d'exécution permet d'introduire la notion de robustesse en ordonnancement. En effet, le problème n'est plus l'optimisation de la durée totale de l'ordonnancement mais l'évaluation de la robustesse de ce dernier face à l'incertitude définie. La question qui se pose ici est : Est-ce que l'ordonnancement est robuste face aux incertitudes modélisées ?

Dans le cadre de ce travail, nous avons considéré que ces durées d'exécution et que les perturbations sur celles-ci étaient des données d'entrées de notre problème.

2.2.2 Caractérisation de la robustesse

Dans le premier chapitre de [5], la robustesse est définie comme suit : *Une solution est robuste si sa performance est peu sensible à l'incertitude des données et aux aléas.* Cette définition, si elle a le mérite d'être consensuelle, reste peu précise dans la mesure où il reste à définir ce que signifie « *être peu sensible à* ». Des métriques, afin de caractériser cette sensibilité, sont donc nécessaires.

Pour définir l'expression de la robustesse, il faut tout d'abord s'intéresser aux éléments suivants:

- Indicateur de robustesse $C(S, I)$:

Un indicateur de robustesse permet de définir l'élément de calcul de la robustesse. L'indicateur de robustesse considéré dans nos travaux est l'écart normé entre la durée totale de l'ordonnancement S sous perturbations I notée $C_{max}(S, I)$ et la durée totale de l'ordonnancement S déterministe, $C_{max}(S, I^{ref})$ avec $I^{ref} = (d_{jkm}^{ref})_{j\in J,\ k\in O,\ m\in M}$:

$$C(S, I) = \frac{C_{max}(S, I) - C_{max}(S, I^{ref})}{C_{max}(S, I^{ref})} \tag{3}$$

- Seuil autorisé :

Ce seuil permet de contraindre l'indicateur de robustesse à respecter les performances souhaitées par le chef d'atelier. Dans notre cas, l'indicateur de robustesse doit respecter un pourcentage $X\%$ de la durée totale de référence de l'ordonnancement. Autrement dit quel est l'écart en pourcentage autorisé entre $C_{max}(S, I)$ et $C_{max}(S, I^{ref})$?

$$\frac{C_{max}(S, I) - C_{max}(S, I^{ref})}{C_{max}(S, I^{ref})} \leq X\% \tag{4}$$

L'expression 4 nous permet d'exprimer le critère de robustesse de la façon suivante :

$$C_{max}(S, I) \leq \tilde{d} \tag{5}$$

avec $\tilde{d}$ qui mesure la durée maximale autorisée fixée à $(X + 100)\% \times C_{max}(S, I^{ref})$.

- Niveau de service p_{lim}:

Le niveau de service p_{lim} fait office de seuil limite que le décideur veut satisfaire, par exemple : le nombre minimum de commandes, le pourcentage d'exécution de l'ordonnancement dans les temps...
Dans le chapitre 5 de [5], l'analyse du niveau de service est définie comme la vérification que la probabilité d'une performance donnée soit supérieure ou égale à un niveau de service souhaité. La modélisation stochastique permet de formuler ce problème de robustesse par l'équation (6). On cherche donc ici à vérifier si la probabilité que la durée totale $C_{max}(S, I)$ d'un ordonnancement respecte une deadline $\tilde{d}$ est supérieure ou égale à un niveau de service défini p_{lim}.

$$Pr(C_{max}(S, I) \leq \tilde{d}) \geq p_{lim} \tag{6}$$

3 Évaluation de la robustesse

Dans cette section, nous allons présenter les deux problèmes de robustesse que nous traitons et l'approche d'évaluation que nous proposons. Cette approche utilise les automates temporisés stochastiques.

3.1 Problèmes de robustesse

L'expression 6 permet de formuler plusieurs problèmes de robustesse, en fonction des paramètres qui sont connus. Dans cet article, nous considérons uniquement deux problèmes d'évaluation de la robustesse :

1. *Analyse du niveau de service:* Ce problème considère que tous les paramètres sont connus et évalue le niveau de service, autrement dit : « Étant donné un ordonnancement S soumis à des incertitudes sur les durées opératoires I, et étant donnée une deadline $\tilde{d}$, quelle est la probabilité que $(C_{max}(S, I) \leq \tilde{d})$ soit vraie ?»

$$Pr(C_{max}(S, I) \leq \tilde{d}) \tag{7}$$

2. *Analyse de sensibilité:* Ce problème considère que l'ordonnancement et les incertitudes sont connus et détermine la date au plus tôt de fin d_{min} de l'ordonnancement, autrement dit : « Étant donné un ordonnancement S soumis à des incertitudes sur les durées opératoires I, quelle est la plus petite deadline d_{min} telle que le niveau de service p_{lim} soit satisfait ? ».

$$Pr(C_{max}(S, I) \leq d_{min}) \geq p_{lim} \tag{8}$$

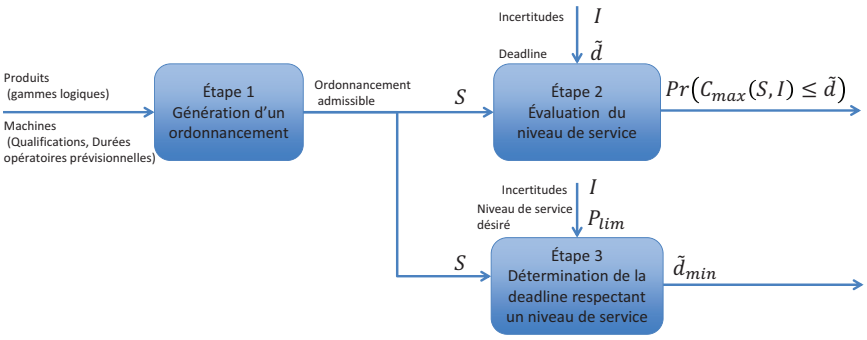

Figure 1: Procédure d'évaluation de la robustesse d'un ordonnancement

3.2 Approche d'évaluation de la robustesse d'un ordonnancement

L'approche proposée pour évaluer les deux problèmes de robustesse précédemment définis, comporte 3 étapes (figure 1).

- La première étape consiste à calculer un ordonnancement déterministe à partir des informations sur l'atelier (qualification des machines, durée d'exécution prévisionnelle de chaque opération sur chaque machine, gammes logiques de production des produits) qui sont considérés à cette étape comme certaines et statiques. L'obtention de cet ordonnancement peut être faite via un expert métier, via une approche classique d'ordonnancement (Métaheuristiques, MILP . . .) ou via une approche SED [4][16][20][15]. Cette étape ne sera pas détaillée dans cet article. Nous ferons l'hypothèse qu'un ensemble d'ordonnancements est connu et doit être évalué du point de vue de la robustesse.
- A partir d'un ordonnancement obtenu dans l'étape 1, des incertitudes sur les durées opératoires I et d'une deadline $\tilde{d}$ acceptée, l'étape 2 évalue le niveau de service.
- A partir d'un ordonnancement obtenu dans l'étape 1, des incertitudes sur les durées opératoires I, la troisième étape détermine la deadline au plus tôt qui satisfait un niveau de service désiré p_{lim}.

3.3 Modélisation

La mise en œuvre de la procédure présentée à la section précédente nécessite d'avoir des modèles qui sont capables de représenter :

- les différents états des ressources
- les gammes logiques de production
- l'affectation et le séquencement des opérations suivant l'ordonnancement
- les incertitudes sur les durées opératoires

Plusieurs outils des SED permettent ce type de modélisation, parmi lesquels on peut citer les automates stochastiques[13][6], les réseaux de Petri stochastiques[3][12] et les réseaux d'automates stochastiques [17][19][18].

3.3.1 Formalisme des automates temporisés stochastiques

Les automates temporisés stochastiques (notés ATS) [6] sont dérivés de la classe des automates temporisés définis par [1] qui partagent des variables et qui sont synchronisés par des événements. Un automate temporisé stochastique est un n-uplet A = (L, V, E, P, H, T, L_m, l_0, v_0), où :

- L est l'ensemble des localités ;
- V est un ensemble de variables entières ;
- E est l'ensemble des événements de synchronisation ;
- P est l'ensemble de probabilités :
  - discrètes sur l'ensemble des transitions (une transition permet d'atteindre différentes localités l_i avec une probabilité p_i où $\sum p_i = 1$);
  - continues sur les variables de l'automate (un tirage aléatoire selon une distribution de probabilités permet de déterminer la condition de franchissement d'une transition).
- H est l'ensemble des horloges.
- T est l'ensemble des transitions $(l,\ e,\ g,\ m,\ l') \in L \times E \times G \times Maj \times L$ où l et l' sont, respectivement, les localités de départ et d'arrivée, G est l'ensemble des gardes (conditions sur les variables de V) et Maj est l'ensemble des mises à jour des évaluations de variables; e, g et m sont facultatifs (une transition peut ne contenir aucun événement de synchronisation et aucune mise à jour par exemple), mais une transition doit contenir au moins un événement ou une garde ;
- $L_m \subseteq L$ est l'ensemble des localités marquées ;
- $l_0 \in L$ est la localité initiale ;
- $v_0 \in V$ est la valeur initiale des variables.

Les états d'un automate sont constitués d'un triplet (l, v, x), où l est la localité active, v est l'ensemble des valuations des variables et x est l'ensemble des valuations des horloges. Un état (l, v, x) peut évoluer :

- suite à un écoulement du temps (des horloges) sous réserve que l'invariant $I(l)$ soit préservé,
- suite au franchissement :
  - d'une transition locale à un automate sous réserve que cette transition ne porte aucun événement de synchronisation,
  - simultané de deux transitions t_p^α, t_q^β d'un couple d'automates (A^α, A^β) avec t_p^α contenant l'étiquette l_p^α ! et t_q^β contenant l'étiquette l_q^β ? telles que $l_p^\alpha = l_q^\beta$, les localités sources de ces transitions étant actives et leurs gardes étant satisfaites,
  - d'une transition probabiliste, sous réserve que sa garde soit satisfaite; dans le cas d'une probabilité discrète, l'évolution est identique à celle d'une chaîne de Markov en temps discret; sur l'exemple de la figure 2, la transition ayant *envoi* comme localité source a une probabilité de 0.99 d'atteindre la localité *Attente* et une probabilité de $1 - 0.99 = 0.01$ d'atteindre la localité *Echec*.

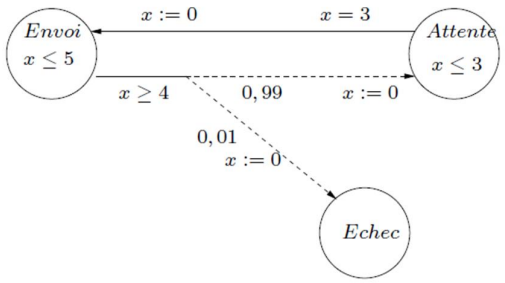

Figure 2: Exemple d'un automate temporisé stochastique [8]

Graphiquement, les probabilités des transitions sont représentées par des arcs en pointillés (figure 2), les invariants sont notés en italique à l'intérieur de la localité, les gardes sont en texte italique et les mises à jour sont texte gras et souligné.

3.3.2 Étape 1 : Modèles de l'ordonnancement

Pour évaluer la robustesse d'un ordonnancement donné, il est nécessaire dans un premier temps de modéliser cet ordonnancement. Il est défini par :

- une affectation de chaque opération à une machine
- le séquencement des opérations sur les machines

De plus, cet ordonnancement étant admissible a priori, il respecte les contraintes de précédence définies par la gamme logique des produits. Nous avons choisi de faire un modèle générique appelé patron de modélisation pour chaque opération o_{jk} du produit j, que nous instancierons en fonction du problème traité. Ce patron représente l'évolution de l'état d'une opération en fonction du déroulement de l'exécution de l'ordonnancement, c'est-à-dire le séquencement des opérations dans la gamme logique et sur les machines. Ce patron est donné en figure 3, il est composé de quatre localités et évolue de la manière suivante:

- Dans la localité ***Idle***, l'opération est en attende d'exécution. Pour que le modèle puisse évoluer, il faut que la garde soit satisfaite. La garde teste si l'opération précédente o_{jk-1} de la gamme logique est exécutée, si l'opération précédente o_{mjk-1} dans la séquence sur la machine m (définie par l'ordonnancement) est exécutée et si le produit est disponible pour subir l'opération o_{jk}. Si la garde est satisfaite, l'exécution peut commencer. Lors du franchissement de la transition, l'horloge x est mise à zéro.
- Une fois la transition franchie, le modèle est dans la localité ***Start Execution***. A partir de cette localité l'exécution de l'opération suit la loi de probabilité de l'incertitude de la durée d'exécution. Une distribution de probabilité est définie sur l'intervalle $[d^{min}_{jkm}, d^{max}_{jkm}]$ et suit une loi de distribution tronquée de paramètre $gamma_x$. Si le paramètre $gamma_x$ est constant, la distribution de probabilité sera une loi exponentielle. En changeant un peu l'automate, par exemple en faisant varier $gamma_x$ est fonction de l'horloge x, la distribution de probabilité pourra être une loi normale, loi Gamma, loi de Bernoulli, loi

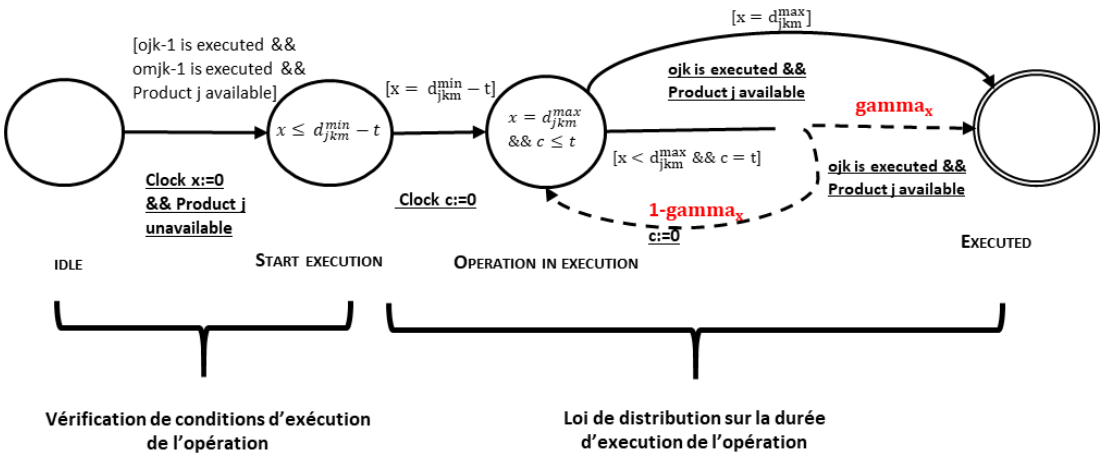

Figure 3: Patron de modélisation de l'opération o_{jk}

de Poisson... Dans le modèle, les horloges x et c permettent respectivement de connaître la valeur de d_{jkm} et le pas de discrétisation noté t.
L'invariant se trouvant dans la localité ***Start Execution*** force le franchissement de la transition lorsque que x atteint $d_{jkm}^{min} - t$.

- En passant à la localité suivante ***Operation In Execution***, l'horloge c est mise à zéro. L'invariant de cette localité permet de rester dans la localité le pas de discrétisation t. Il contraint aussi l'horloge x à être inférieure à d_{jkm}^{max}. Dès que $c == t$, deux choix probabilistes sont possibles:

  1. Avec la probabilité $gamma_x$, l'exécution de l'opération est terminée, le modèle franchit la transition vers la localité ***Executed*** et met à jour l'état de l'opération o_{jk}.
  2. Avec la probabilité $1 - gamma_x$, l'opération revient dans la localité ***Operation In Execution*** et c est remise à zéro pour avancer d'un autre pas t.

- A partir de la localité ***Operation In Execution***, quand l'horloge x atteint d_{jkm}^{max}, l'invariant n'est plus satisfait, l'évolution vers la localité ***Executed*** est obligatoire et l'état de l'opération o_{jk} est mis à jour. Pour permettre l'enregistrement du temps total d'ordonnancement, une horloge globale à tous les modèles d'opération notée clk est modélisée. Cette horloge correspond à $C_{max}(S, I)$, la durée totale de l'ordonnancement.

3.4 Étapes 2 et 3: Évaluation de la robustesse

A partir du modèle présenté précédemment, nous proposons de mettre en œuvre les étapes 2 et 3 de la procédure présentée sur la figure 1. L'idée est, en utilisant les outils de model-checking, d'évaluer la probabilité qu'une propriété soit satisfaite.

Pour l'évaluation du niveau de service (étape 2 de la procédure), la question posée au model-checker peut être décrite de la façon suivante : quelle est la probabilité de l'ensemble des chemins qui, en une durée inférieure à la deadline donnée $\tilde{d}$, atteignent un état global où

tous les modèles d'opération sont dans la localité marquée ***Executed*** ? Formellement, cette question s'écrit à l'aide de la formule Probabilistic Computation Tree Logic (PCTL) suivante :

$$P =?[F \leq \tilde{d} \text{ "All operations } o_{jk} \text{ are Executed"}] \tag{9}$$

L'expression 9 permet d'analyser le niveau de service exprimé dans 7, en effet, l'opérateur P fait référence à la probabilité Pr et le $=?$ exprime le fait de calculer cette probabilité. Le $C_{max}(S, I)$ est représenté par la valeur de l'horloge globale clk quand toutes les opérations sont arrivées dans la localité ***Executed***. La vérification de $C_{max}(S, I) \leq \tilde{d}$ est exprimée par $F \leq \tilde{d}$ "*All operations* o_{jk} *are Executed*".

Dans l'étape 3, il faut déterminer la plus petite valeur de $\tilde{d}$ notée d_{min} pour laquelle la probabilité que tous les modèles d'opérations atteignent la localité ***Executed*** en moins de $\tilde{d}$ unités de temps est supérieure à un niveau de service fixé p_{lim}. Cela revient à trouver la plus petite valeur d_{min} pour laquelle la propriété suivante est vérifiée :

$$P_{\geq p_{lim}}[F \leq d_{min} \text{ "All operations } o_{jk} \text{ are Executed"}] \tag{10}$$

L'expression 10 permet de faire l'analyse de sensibilité exprimée dans 8. En effet, l'expression $\leq p_{lim}$ permet de comparer la probabilité calculée avec la valeur du niveau de service. Afin de déterminer la date d_{min}, un algorithme de recherche par dichotomie peut être utilisé sur l'intervalle $[0, C_{max}(S, I_{max})]$ où $C_{max}(S, I_{max})$ est la durée de l'ordonnancement S en considérant les durées maximales d'exécution d_{jkm}^{max}, relancer le model-checker autant de fois que nécessaire en mettant à jour la valeur de $\tilde{d}$ suivant que la propriété est vérifiée ou non.

Il s'agit maintenant de choisir le model-checker qui va permettre de vérifier ces propriétés. Parmi les outils de vérification automatique de systèmes probabilistes on peut distinguer deux méthodes : le model-checking numérique et le model-checking statistique. Le model-checking numérique consiste, en utilisant des méthodes numériques basées sur du calcul matriciel, de s'approcher aussi près que souhaité de la valeur exacte d'une probabilité. Le model-checking statistique pour sa part se base sur l'échantillonnage d'exécutions afin d'estimer par des méthodes de type Monte Carlo les quantités souhaitées. Si l'avantage du premier réside dans la précision des résultats obtenus (sans l'incertitude induite par l'utilisation de méthodes statistiques), le deuxième permet d'éviter l'explosion combinatoire car il ne requiert pas la construction de l'espace d'états du système, et autorise également des modèles plus riches (en termes d'évolution des variables ou de types de distributions considérées) qui rendraient le modèle indécidable dans le cadre du model-checking probabiliste [2].

Pour répondre à la problématique en prenant en compte à la fois les aspects temporisés inhérents au problème d'ordonnancement et la variabilité du système, nous avions besoin d'un outil tolérant à la fois des actions temporisées déterministes mais aussi introduire un aspect stochastique pour certains autres délais (comme le traitement des durées d'exécution des opérations sur les machines). Nous avons donc porté notre choix sur deux outils ayant ces caractéristiques : UppAal SMC et PRISM. UppAal SMC version 4.1.14 [7] est un outil de model-checking qui prend en entrée des modèles en ATS et est conçu pour donner une sémantique probabiliste à nos systèmes temporisés. L'outil PRISM [14], plus répandu pour la vérification des systèmes probabilistes, permet de faire du model-checking numérique et statistique et utilise en entrée des modèles définis par processus de décision markovien (MDP).

L'approche statistique est basée sur la méthode de Monte-Carlo, le model-checker simule un grand nombre de fois le modèle à évaluer, jusqu'à obtenir un ensemble d'échantillons d'observations suffisants pour estimer la probabilité. Pour déterminer le nombre d'échantillon, il est nécessaire de définir trois paramètres : l'intervalle de confiance, la probabilité d'être dans

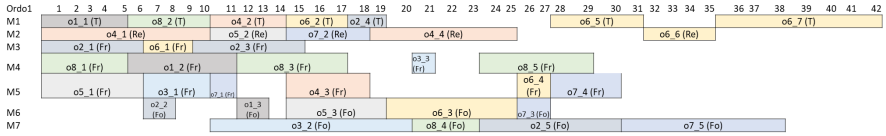

Figure 4: Ordonnancement possible (S1) en 42 unités de temps

cet intervalle et le nombre de simulations à effectuer. UppAal détermine automatiquement le nombre de simulation en fonction des deux autres paramètres alors que PRISM demande de fixer les trois paramètres.

4 Expérimentation

Nous proposons d'expérimenter notre approche sur un exemple académique [9] et adapté pour faire apparaître des incertitudes sur les temps d'exécution. L'objectif de cet exemple est d'ordonnancer huit produits donnés, dans un atelier contenant un tour, une rectifieuse, trois fraiseuses et deux fours. Les qualifications des machines et leurs durées opératoires minimales et maximales sont données dans la table 1. Nous rappelons que l'objectif de cet article est de montrer la faisabilité d'utiliser les SED pour l'évaluation de la robustesse. En l'absence de mesures précises de la variabilité des durées opératoires, nous avons choisi arbitrairement une loi exponentielle pour représenter l'incertitude sur la durée d'exécution.

Prod.	Opéra.	Ressource	d_{jkm}^{min}	d_{jkm}^{max}	Prod.	Opéra.	Ressource	d_{jkm}^{min}	d_{jkm}^{max}
1	$o1\_1$	Tour	4	6	6	$o6\_1$	Fraiseuse	2	4
1	$o1\_2$	Fraiseuse	4	7	6	$o6\_2$	Tour	2	4
1	$o1\_3$	Four	1	3	6	$o6\_3$	Four	6	9
2	$o2\_1$	Fraiseuse	4	7	6	$o6\_4$	Fraiseuse	1	3
2	$o2\_2$	Four	1	3	6	$o6\_5$	Tour	3	5
2	$o2\_3$	Fraiseuse	4	7	6	$o6\_6$	Rectifieuse	3	5
2	$o2\_4$	Tour	1	3	6	$o6\_7$	Tour	5	8
2	$o2\_5$	Four	5	8	7	$o7\_1$	Fraiseuse	1	3
3	$o3\_1$	Fraiseuse	3	5	7	$o7\_2$	Rectifieuse	3	5
3	$o3\_2$	Four	8	11	7	$o7\_3$	Four	1	3
3	$o3\_3$	Fraiseuse	1	3	7	$o7\_4$	Fraiseuse	2	4
4	$o4\_1$	Rectifieuse	8	11	7	$o7\_5$	Four	6	9
4	$o4\_2$	Tour	3	5	8	$o8\_1$	Fraiseuse	4	6
4	$o4\_3$	Fraiseuse	3	5	8	$o8\_2$	Tour	4	6
4	$o4\_4$	Rectifieuse	6	8	8	$o8\_3$	Fraiseuse	4	7
5	$o5\_1$	Fraiseuse	4	7	8	$o8\_4$	Four	2	4
5	$o5\_2$	Rectifieuse	4	6	8	$o8\_5$	Fraiseuse	4	7
5	$o5\_3$	Four	4	6					

Table 1: Gamme logique des produits et caractéristiques des machines

A partir de nos travaux [10], les ordonnancements des figures 4, 5 et 6 ont été obtenus et ils ont une durée totale du même ordre de grandeur. Afin d'aider le décideur dans le choix de l'ordonnancement à lancer dans l'atelier, nous allons évaluer dans un premier temps la robustesse en sachant qu'une deadline $\tilde{d}$ est acceptée (avec $\tilde{d} = 110\% \times C_{max}(S, I^{ref})$) et dans un second temps, sachant que le décideur souhaite un niveau de service de 85% ($p_{lim} = 85\%$), nous déterminerons la durée totale la plus courte permettant d'assurer ce niveau de service en utilisant l'algorithme de dichotomie sur les intervalles $[0, 52]$, $[0, 41]$, et $[0, 43]$ où 52, 41, et 43 représentent respectivement les durées totales maximales pour les 3 ordonnancements (dans le cas où la durée de chaque opération prend le temps maximal autorisé).

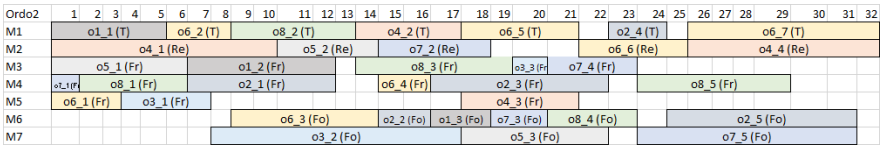

Figure 5: Ordonnancement possible (S2) en 32 unités de temps

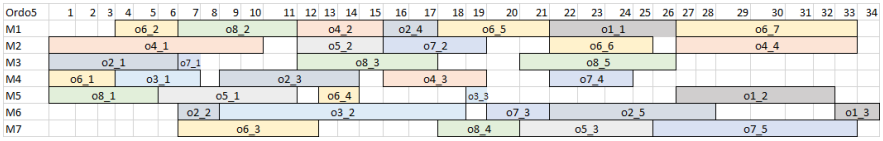

Figure 6: Ordonnancement possible (S3) en 34 unités de temps

Pour chacun de ces ordonnancements, nous retenons seulement leur durée $C_{max}(S, I^{ref})$, l'affectation et le séquencement des opérations sur les machines. Le patron du modèle d'opération de la figure 3 est instancié sous UppAal SMC et PRISM en tenant compte du séquencement et de l'affectation des opérations sur les machines et de la gamme logique des produits. Par exemple pour l'opération $o5\_2$ de l'ordonnancement de la figure 4, il faut vérifier si l'opération $o5\_1$ de la gamme logique du produit 5 et l'opération $o4\_1$ précédant l'opération $o5\_2$ sur la rectifieuse sont exécutées.

Dans le logiciel UppAal, nous avons synchronisé l'évolution des modèles en utilisant des canaux de communications et dans le logiciel PRISM, nous avons utilisé des événements synchronisants.

Afin d'évaluer la robustesse des trois ordonnancements par rapport à la date de fin obligatoire à $\tilde{d}$, la propriété (9) a été évaluée et il en est ressorti l'analyse du niveau de service de la figure 7.a, où les ordonnancements 1 et 3 sont les plus robustes.

Pour définir la date de fin au plus tôt qui permet un niveau de service de 85%, la propriété (10) a été vérifiée et cette fois, les ordonnancement 2 et 3 ont des deadlines très proches pour satisfaire le niveau de service de 85% (figure 7.b). Suite à ces évaluations, il semble que l'ordonnancement 3 est le meilleur choix à faire au vu des incertitudes sur les temps d'exécution, alors que sur le seul critère du temps d'exécution global, l'ordonnancement 2 aurait été choisi.

	Analyse du niveau de service			Analyse de sensibilité		
	Résultat	Nb sim	T. C.	Résultat	Nb sim	T. C.
N°1 : 42UT	$\tilde{d} = 46$: $P \in [0.978, 0.9984]$	$535 sim.$	$2.64s$	$d_{min} = 43$ $P \in [0.888, 0.9086]$	$5017 sim.$	$24.84s$
N°2: 32UT	$\tilde{d} = 35$: $P \in [0.9434, 0.9634]$	$2468 sim.$	$12.81s$	$d_{min} = 34$ $P \in [0.8645, 0.8845]$	$6028 sim.$	$31.53s$
N°3: 34UT	$\tilde{d} = 37$: $P \in [0.9716, 0.9916]$	$983 sim.$	$4.92s$	$d_{min} = 35$ $P \in [0.9052, 0.9252]$	$4280 sim.$	$21.48s$

Table 2: Évaluation des ordonnancements en SMC avec le model-checker UppAal

La table 2 présente les résultats des évaluations avec le model-checker UppAal ainsi que le temps de calculs (noté T.C. dans le tableau), le nombre de simulations, et l'intervalle de

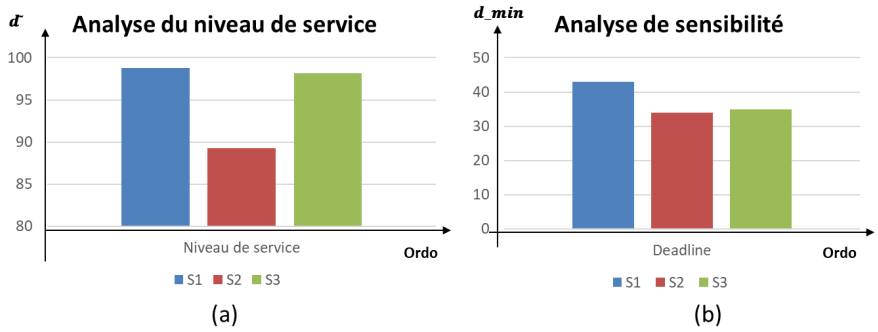

Figure 7: Résultats de l'évaluation des ordonnancements

précision. Les résultats obtenus avec le model-checker PRISM sont similaires.

5 Conclusion

Cet article a proposé une approche basée sur les modèles de SED qui permet d'évaluer la robustesse d'un ordonnancement face à des incertitudes sur les temps opératoires. Cette approche est basée sur des automates temporisés stochastiques pour modéliser l'évolution des opérations et des machines et sur du model-checking stochastique pour évaluer formellement la robustesse d'un ordonnancement donné. Les modèles proposés pour évaluer la robustesse sont génériques et indépendants du type d'atelier alors que dans les approches classiques, ce sont des algorithmes de calcul dédiés qui doivent être mis en place en fonction du type d'atelier. Dans cet article, nous avons évalué le niveau de service en fonction d'une durée totale d'ordonnancement souhaitée et déterminé la durée la plus courte en fonction d'un niveau de service donné.

Dans la suite de nos travaux, nous souhaitons étendre notre approche afin qu'elle permette de prendre en compte d'autres types de distribution pour les modèles d'incertitudes et d'autres types de perturbation tels que des aléas (par exemple des pannes machine). Il pourrait être intéressant de modifier les algorithmes de recherche d'atteignabilité utilisés dans PRISM et UppAal pour améliorer leur performance en fonction de nos besoins d'évaluation. Dans cet article, nous avons considéré l'ordonnancement connu et il pourrait être intéressant, par la suite, de proposer une approche qui calcule un ordonnancement en fonction de perturbations connues et d'un niveau de robustesse souhaitée.

References

[1] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[2] P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin. Hasl an expressive language for statistical verification of stochastic models. *Poceeding 5th, International Conference Value TOOLS*, 2011.

[3] 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.

[4] G. Behrmann, E. Brinksma, M. Hendriks, and A. Mader. Production scheduling by reachability analysis - a case study. *International Parallel and Distributed Processind Symposium*, 2005.

[5] Jean-Charles Billaut, Aziz Moukrim, and Eric Sanlaville. *Flexibility and Robustness in Scheduling*. ISTE, 2010.

[6] C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems, second edition*. Springer, 2008.

[7] Alexandre David, Kim G Larsen, Axel Legay, Marius Mikučionis, and Danny Bøgsted Poulsen. Uppaal smc tutorial. *International Journal on Software Tools for Technology Transfer*, 17(4):397–415, 2015.

[8] Conrado Daws, Marta Kwiatkowska, and Gethin Norman. Automatic verification of the ieee 1394 root contention protocol with kronos and prism. *International Journal on Software Tools for Technology Transfer (STTT)*, 5(2):221–236, 2004.

[9] Vincent Giard. *Gestion de la production et des flux: avec CD livre électronique+ Logiciels+ Animations*. Economica, 2003.

[10] Sara Himmiche, Pascale Marangé, Alexis Aubry, and Jean-François Pétin. Ordonnancement d'ateliers à partir de patrons de modélisation basés sur des automates communicants. In *11th International Conference on Modeling, Optimization and Simulation*, Montréal, Canada, August 2016.

[11] A. Kobetski and M. Fabian. Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming. *Journal of Discrete-Event Dynamic Systems: Theory and Applications*, 19(3):287–315, 2009.

[12] P.S. Kritzinger and F. Bause. *Stochastic Petri nets - An Introduction to the theory*. Vieweg+Teubner, 2002.

[13] M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design*, 2006.

[14] Marta Kwiatkowska, Gethin Norman, and David Parker. Prism: Probabilistic symbolic model checker. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 200–204. Springer, 2002.

[15] Pascale Marangé, Jean-François Pétin, Antoine Manceaux, and David Gouyon. Contribution à la reconfiguration des systèmes de production : ordonnancement par recherche d'atteignabilité. *Journal Européen des Systèmes Automatisés (JESA)*, 45(1/3):45–60, November 2011.

[16] S. Panek, S. Engell, and O. Stursberg. Scheduling and planning with timed automata. In *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, pages 1973–1978. Elsevier, 2006.

[17] B. Plateau and K. Atif. Stochastic automata networks for modelling parallel systems. *Reo Project*, 1991.

[18] Correa De Sales. *Réseaux d'automates stochastiques: Génération de l'espace d'états atteignables et multiplication vecteur-descripteur pour une sémantique en temps discret*. PhD thesis, Institut polytechnique de Grenoble, 2009.

[19] W. J. Stewart, K. Atif, and B. Plateau. The numerical solution of stochastic automata networks. *European Journal of Operational Research*, 1995.

[20] S. Subbiah and S. Engell. Short-term scheduling of multi-product batch plants with sequence-dependent changeovers using timed automata models. In *20th European Symposium on Computer Aided Process Engineering*, number 28, pages 1201–1206. Elsevier, 2010.