



**HAL**  
open science

# Reconfiguration dynamique d'une ligne d'assemblage par recherche d'atteignabilité sur un réseau d'automates communicants

Antoine Manceaux, Hind Bril El-Haouzi, André Thomas, Jean-François Pétin,  
Thierry Munier

## ► To cite this version:

Antoine Manceaux, Hind Bril El-Haouzi, André Thomas, Jean-François Pétin, Thierry Munier. Reconfiguration dynamique d'une ligne d'assemblage par recherche d'atteignabilité sur un réseau d'automates communicants. 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, MOSIM'14, Nov 2014, Nancy, France. hal-01094668

**HAL Id: hal-01094668**

**<https://hal.science/hal-01094668>**

Submitted on 12 Dec 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# RECONFIGURATION DYNAMIQUE D'UNE LIGNE D'ASSEMBLAGE PAR RECHERCHE D'ATTEIGNABILITÉ SUR UN RÉSEAU D'AUTOMATES COMMUNICANTS

A. MANCEAUX<sup>1,2,3</sup>, H. BRIL EL-HAOUZI<sup>1,2</sup>, A. THOMAS<sup>1,2</sup>, J.F. PÉTIN<sup>1,2</sup>, T. Munier<sup>3</sup>

<sup>1</sup>Université de Lorraine, CRAN - Campus Sciences BP  
70239 - 54506 VANDOEUVRE Cedex  
antoine.manceaux@univ-lorraine.fr, hind.el-  
haouzi@univ-lorraine.fr, andre.thomas@univ-  
lorraine.fr, jean-francois.petin@univ-lorraine.fr

<sup>2</sup>CNRS, CRAN, UMR 7039, France  
<sup>3</sup>TRANE SAS, rue des Amériques, 88190 Golbey,  
France  
Antoine\_Manceaux@trane.com

**RÉSUMÉ :** Cet article propose une méthode s'appuyant sur la recherche d'atteignabilité sur un réseau d'automates communicants. Celle-ci est utilisée pour rééquilibrer dynamiquement une ligne d'assemblage lorsqu'une perturbation survient. Le but de la méthode est de réallouer dynamiquement les tâches sur les postes de travail de cette ligne. Un réseau d'automates est utilisé pour modéliser le système et une solution est obtenue par la trace fournie par le logiciel de model-checking effectuant la recherche d'atteignabilité. La méthode est illustrée sur un cas d'étude.

**MOTS-CLÉS :** Ligne d'assemblage, Rééquilibrage, Dynamique, Reconfiguration, Système à évènements discrets, Recherche d'atteignabilité.

## 1 INTRODUCTION LE PROBLÈME TRANE, UN PROBLÈME DE RÉÉQUILIBRAGE DE LIGNE

La société Trane, qui nous sert de cas d'étude, est une industrie produisant des appareils de conditionnement d'air selon une stratégie de type personnalisation de masse. Comme beaucoup d'industries de ce type, ses lignes d'assemblages sont organisées en flux tirés et en lignes de production dédiées à une famille de produits. Pour ces types de système de production, un des problèmes à résoudre est le fait d'allouer correctement les tâches à réaliser sur le produit sur les différents postes de travail qui constituent la ligne. Ce problème étudié depuis de nombreuses années par les industriels et par les chercheurs en recherche opérationnelle se nomme l'équilibrage de ligne d'assemblage (ALBP pour Assembly Line Balancing Problem).

Cet article démontre la pertinence de l'utilisation de la recherche d'atteignabilité sur un réseau d'automates communicants pour le rééquilibrage réactif d'une ligne d'assemblage comportant des opérations manuelles. Après la présentation du contexte dans la section 2, une description formelle du problème est présentée en section 3. La méthode est ensuite présentée et expliquée dans la section 4. L'application industrielle pour la société Trane est le thème de la section 5. Enfin la section 6 conclut et présente les perspectives et les travaux futurs.

## 2 TRAVAUX RELATIFS A L'ÉQUILIBRAGE DE LIGNES ET PROBLEMATIQUE

Divers travaux bibliographiques tels que (Scholl et Becker, 2006), (Boysen *et al.*, 2008), (Becker et Scholl, 2006), et (Battaia et Dolgui, 2013) ont conduit à proposer différentes méthodes et typologies relatives à ce problème. Ainsi il existe quatre types de problèmes dit simples et pour lesquels des études comparatives des méthodes existent. Pour les autres problèmes dits « généraux », (Boysen *et al.*, 2007) ont proposé une caractérisation des problèmes se basant sur les n-uplets  $[\alpha | \beta | \gamma]$  de (Graham *et al.*, 1979) utilisés en ordonnancement :

- un ensemble  $\alpha$  décrivant les tâches et la forme du graphe de précédence ;
- un ensemble  $\beta$  détaillant les caractéristiques de la ligne et des postes de travail ;
- un ensemble  $\gamma$  listant les objectifs et les contraintes.

(Scholl, 1999) définit trois niveaux d'équilibrage de ligne en fonction de l'horizon de la décision et du type de changement nécessaire :

- lors de l'installation de la ligne (horizon de 2 à 5 ans) (dans une nouvelle usine ou une nouvelle ligne dans une usine déjà existante) ;
- lors d'une réingénierie d'une ligne, comme par exemple dans (Grangeon *et al.*, 2011) ou (Maksoud *et al.*, 2013) (horizon de 6 mois à 2 ans). Modification d'une ligne d'assemblage existante pour intégrer un nouveau produit par exemple. Cela conduit à ajouter

des contraintes. Ainsi des coûts supplémentaires peuvent apparaître dans le modèle d'optimisation.

- enfin pour un rééquilibrage calculé, par exemple, pour faire face à un changement des dimensions à produire (horizon 1 mois à 1 an).

Pour les horizons plus faibles (inférieur à 1 mois), l'ajustement de la production est réalisé en modifiant le séquençement de la ligne avec 2 échelles de temps : le « master scheduling » (pour un horizon d'une semaine par exemple) ou le séquençement journalier. Par exemple, le sujet du challenge de la ROADEF en 2005 consistait à trouver un algorithme pour trouver le séquençement de la journée de production d'une usine de production d'automobiles comme par exemple celui de (Estellon *et al.*, 2008).

De plus, ce re-séquencement peut aussi avoir lieu à une échelle locale et réactive, par exemple en inversant l'ordre de deux produits sur un poste lorsqu'un aléa comme une rupture d'approvisionnement survient.

Dans le cas d'application de cet article, (décrit en section 5), la grande taille des produits fabriqués et la conception de la ligne nous interdisent les re-séquencements. Lorsqu'un aléa apparaît, il nous faut modifier localement l'allocation initiale des tâches afin d'adapter l'équilibrage. L'équilibrage initial tenait compte de la ligne entière, mais évidemment il ne pouvait pas tenir compte des perturbations locales. Dans notre cas, le temps de réaction étant plus important que l'optimalité, nous nous contenterons de l'obtention d'une solution acceptable.

Dans le but de détecter les perturbations, puis de les appliquer (par message aux opérateurs), le système de résolution doit connaître l'état courant de la ligne d'assemblage. Ceci sera obtenu par le fait que les produits captent, portent et diffusent des informations grâce à des technologies de type Auto-ID. Nous nous plaçons, de ce fait, dans le contexte des systèmes manufacturiers intelligents (intelligent manufacturing systems, IMS).

Ainsi nous pouvons positionner notre problématique de rééquilibrage dynamique dans un contexte plus large comme présenté sur la figure 1. La production est réalisée avec un équilibrage calculé de manière centralisée et prédictive qui peut être recalculé lors d'une reconception, ou de l'arrivée d'un nouveau produit.

Lorsqu'un aléa survient (que cela génère une avance ou un retard), un rééquilibrage dynamique, tel que présenté dans ce papier, est calculé. Cette solution, si elle existe, doit alors être appliquée sur la ligne en déplaçant certaines tâches.

Enfin, avant de relancer la production, l'incident peut être enregistré afin de détecter des problèmes récurrents qui démontreraient que l'équilibrage initial n'est plus

adapté aux conditions actuelles et qu'il doit être recalculé de manière centralisée.

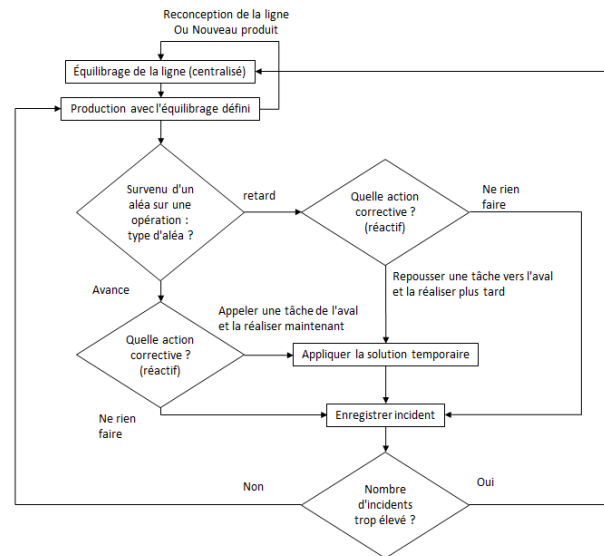


Figure 1 : Rééquilibrage dynamique par rapport à l'équilibrage classique

### 3 FORMALISATION DU PROBLÈME

#### 3.1 Données

- $T = \{t_i, i \leq T_{max} \in \mathbb{N}\}$  est l'ensemble des tâches (avec  $t_i$  l'identifiant de la tâche et  $T_{max}$  le nombre de tâches)
- $W = \{w_i, i \leq W_{max} \in \mathbb{N}\}$  est l'ensemble des postes de travail de la ligne (avec  $w_i$  l'identifiant du poste et  $max$  le nombre de postes)
- $E = \{e_i, i \leq E_{max} \in \mathbb{N}\}$  est l'ensemble des évènements de synchronisation, c'est-à-dire les instants où les produits changent de postes sur la ligne (avec  $E_{max}$  le nombre d'évènements pendant la période étudiée)
- $Dt = \{dt_i, \in \mathbb{N}, i \leq T_{max} \in \mathbb{N}\}$  est l'ensemble des durées des tâches
- $Dw = \{dw_i, \in \mathbb{N}, i \leq W_{max} \in \mathbb{N}\}$  est l'ensemble des durées des postes de travail qui sont définies par les sommes des durées des tâches qui sont allouées à chaque poste

#### 3.2 Contraintes et objectifs

L'allocation est donnée par la fonction surjective  $A_w : T \rightarrow W$ . Pour cela, elle associe à chaque tâche  $t_i$ , le poste de travail  $w_k$  sur laquelle celle-ci est allouée :  $A(t_i) = w_k$ . Une tâche doit être allouée à un et seulement un poste de travail.

#### 3.3 Contraintes

##### 3.3.1 Takt time (CI)

Une allocation d'une tâche sur un poste de travail est possible uniquement si la charge de travail restant sur un poste le permet, c'est-à-dire si la durée du travail sur ce

poste n'exécède pas le takt time (l'objectif est donc que la somme des durées des tâches soit le plus près possible du takt time, et cela pour chacune des opérations (poste de travail) de la ligne). Notons que l'écart de cette somme au takt time est nommé  $\varepsilon$ . Dans ce papier, nous privilégions la rapidité de résolution à l'optimalité, c'est pourquoi  $\varepsilon$  n'atteindra pas forcément sa valeur minimale.

### 3.3.2 Précédence (C2)

$P$  est la matrice de précédence. Pour un couple de tâches  $(t_i, t_j) \in T^2$ ,  $P_{ij}$  vaut "1" si  $t_i$  doit précéder  $t_j$ .

$$\forall t_i \in T, \forall t_j \in T, P_{ij} = \begin{cases} 1 & \text{si } t_i \text{ précède } t_j \\ 0 & \text{sinon} \end{cases}$$

## 3.4 Objectif

La reconfiguration (le rééquilibrage de la ligne d'assemblage) consiste à trouver une solution acceptable, c'est-à-dire une nouvelle allocation  $A_w$  qui respecte les contraintes précédemment citées.

Lors de la recherche d'un équilibrage optimal, le but est de minimiser  $\varepsilon$ . Dans cet article, notre objectif est de trouver une solution possible qui permet de poursuivre la production sans perte de capacité.

## 3.5 Précédents travaux avec cette méthode

Comme écrit précédemment, des méthodes de rééquilibrage ont déjà été proposées, mais elles traitent plus souvent de réingénierie plus que de recherches de solutions purement réactives. Le problème de rééquilibrage peut être traité, une fois les contraintes formalisées, par les méthodes traditionnelles de la recherche opérationnelle. De par la complexité du problème due aux nombreuses contraintes à prendre en compte (Falkenauer, 2005), les méthodes exactes sont souvent délaissées au profit des métaheuristiques comme par exemple dans (Grangeon *et al.*, 2011). Ces méthodes sont beaucoup utilisées pour les créations ou les modifications de lignes. Néanmoins, pour une réactivité dynamique, il nous faut obtenir la solution dans un temps compatible avec l'échelle de temps du système. Pour cela, il faut non seulement une résolution rapide, mais aussi une initialisation aisée.

Une fois les contraintes définies, nous pouvons appliquer une approche basée sur les systèmes à événements discrets (SED) ayant émergé pour modéliser et résoudre les problèmes de séquençement. L'efficacité des automates temporisés et de la recherche d'atteignabilité dans ces cas-là a été démontrée plus particulièrement par (Behrmann *et al.*, 2005) et (Subbiah et Engell, 2010). L'idée est d'utiliser la recherche d'atteignabilité associée aux outils de vérification de modèle (model-checking) (Clarke et Emerson, 1982) pour trouver un chemin possible permettant d'atteindre un certain état du système où toutes les tâches seraient réallouées et la ligne rééquilibrée. La solution trouvée est alors constituée par les

valeurs d'allocation enregistrées pendant l'exécution de la trace conduisant de l'état initial à l'état souhaité.

En plus de la rapidité pour trouver une solution acceptable (pas forcément optimale), l'avantage principal de cette approche avec les automates est sa modularité et sa modélisation facilement paramétrable.

## 4 LA RECHERCHE D'ATTEIGNABILITÉ POUR LE RÉÉQUILIBRAGE

### 4.1 Principe de l'approche

Notre approche est basée sur la modélisation des tâches et des postes de travail où elles doivent être affectées à l'aide d'un ensemble d'automates communicants, soit :

- Le modèle des tâches TM ;
- Le modèle des postes de travail WM ;
- La synchronisation entre les modèles des tâches et des postes est assurée par un mécanisme d'appel/réponse utilisant des automates communicants utilisés par (Lematre, 2013) et (Marangé *et al.*, 2011).

### 4.2 Formalisme utilisé

Les automates communicants sont une sous-classe des automates temporisés définis par (Alur et Dill, 1994). Ceux-là partagent des variables et sont synchronisés par l'ajout de labels / d'étiquettes sur les transitions.

Un automate communicant  $A$  est défini par un  $N$ -uplet  $A = (D, X, L, T, Q_m, q_0, v_0)$ , avec :

- $Q$  est un ensemble fini de localités ;
- $X$  est un ensemble fini de variables entières;
- $L$  est un ensemble d'étiquettes composé de 3 ensembles disjoints :
  - $L_r$  l'ensemble des étiquettes de réception notées *label!*;
  - $L_e$  l'ensemble des étiquettes d'émission notées *label!*;
  - $L_l$  l'ensemble des étiquettes locales.
- $T$  est un ensemble de transitions  $(q, l, g, m, q')$   $Q \times L \times G \times M \times Q$  avec  $G$  l'ensemble des gardes (conditions sur les variables de  $X$ ) et  $M$  est l'ensemble des mises à jour sur les valuations des variables;  $l, g$  et  $m$  sont optionnels (une transition peut contenir aucune étiquette de synchronisation et aucune mise à jour par exemple) mais une transition doit contenir au moins une étiquette ou une garde;
- $q_0 \in Q$  est la localité initiale;
- $v_0 : X \leftarrow \mathbb{N}$  est la valuation initiale des variables

L'état d'un automate, à un certain instant, est défini par la localité active à laquelle on associe les valeurs des variables au même instant.

Un réseau  $NA = A1 \parallel A2 \parallel \dots \parallel An$  avec  $n$  automates ( $n \in \mathbb{N}^*$ ) est défini par  $NA = (Q, X, L, T, Q_m, q_0, v_0)$ , avec :

- $Q \subseteq Q^1 \times Q^2 \times \dots \times Q^n$
- $X = X^1 \cup X^2 \cup \dots \cup X^n$
- $L = L^1 \cup L^2 \cup \dots \cup L^n$
- $T \subseteq Q \times L \times G \times M \times Q$ , avec  $G$  l'ensemble des gardes (conditions sur les variables de  $X$ ) et  $M$  l'ensemble des mises à jour des valuations des variables de  $X$
- $Q_m = Q_m^1 \times Q_m^2 \times \dots \times Q_m^n$
- $q_0 = q_0^1 \times q_0^2 \times \dots \times q_0^n$
- $v_0 : X \leftarrow \mathbb{N}$  est la valuation initiale des variables.

Un état du réseau est défini par un couple  $(q, v)$  avec  $q \in Q$  et  $v \in X$ . Deux types d'évolutions du réseau d'automates peuvent avoir lieu :  $NA(q, v) \xrightarrow{t} (q', v_0)$  peut avoir lieu :

- une évolution est le franchissement d'une transition dans un seul automate si l'étiquette est locale et si la garde associée est vérifiée ;
- deux transitions,  $t_k^\gamma, t_m^\beta$ , d'un couple d'automates,  $(A^\gamma, A^\beta)$ , avec  $t_k^\gamma$  contenant l'étiquette d'émission,  $l_k^\gamma \in L^\gamma$ , notée  $l_k^\gamma!$  et  $t_m^\beta$  contenant l'étiquette de réception,  $l_m^\beta \in L^\beta$ , notée  $l_m^\beta?$ , telles que  $l_k^\gamma = l_m^\beta$  sont franchies simultanément (les gardes étaient satisfaites).

Finalement, une trace de  $A$  est une séquence d'évolutions depuis l'état initial :

$$(s_0, v_0) \xrightarrow{t_1} (s_1, v_1) \xrightarrow{t_2} (s_2, v_2) \dots \xrightarrow{t_n} (s_n, v_n).$$

Nous pouvons noter que des transitions simultanées ne sont possibles que lorsque deux transitions de deux automates distincts sont considérées. Il est impossible d'avoir un mécanisme de diffusion (broadcast) qui implique plus de deux automates. La figure 2 montre un exemple de l'évolution d'automates temporisés : à l'instant  $t$ , la localité 3 de l'automate  $A^1$  est active et la localité 6 de l'automate  $A^2$  aussi. Lorsque la transition  $t_{13}$  est franchissable, le message 3 est émis et est simultanément reçu par la transition  $t_{26}$  qui devient elle aussi franchissable.

Les conventions de notations sont les suivantes : les localités initiales sont indiquées par un arc source, le nom des localités est inscrit en gras, les noms des étiquettes de synchronisation sont en italique et suivi par le symbole « ! » pour l'émission et par le symbole « ? » pour la réception, les mises à jour des variables sont soulignées et les gardes sont écrites entre crochets.

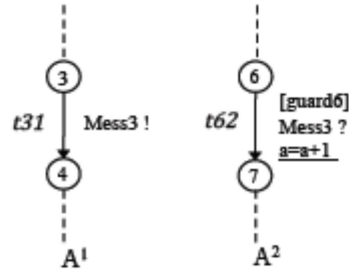


Figure 2 : Exemple de l'évolution synchrone de deux automates communicants

### 4.3 Description de l'approche proposée

#### 4.3.1 Modèle générique d'une tâche

Le modèle générique TM d'une tâche  $t_i$  est défini par un automate avec trois localités et présenté ci-dessous et sur la figure 3.

- Dans la première localité, la tâche  $t_i$  est en attente d'une allocation sur un poste de travail  $w_k$ . La transition qui peut être franchie correspond à l'émission d'une demande d'allocation envoyée à un poste de travail  $w_k$ .
- Une fois cette transition franchie, le modèle est en attente d'une réponse envoyée par le modèle de poste de travail (voir section suivante) qui peut être :
  - un refus et dans ce cas la tâche retourne à sa localité initiale, prête pour une nouvelle demande (Le délai entre une réponse et une nouvelle demande est instantanée, c'est la vérification des contraintes qui utilise du temps de calcul),
  - une acceptation et dans ce cas le modèle passe dans la dernière localité.
- La dernière localité du modèle représente une tâche allouée qui ne peut pas faire de nouvelle demande (unicité de l'allocation).

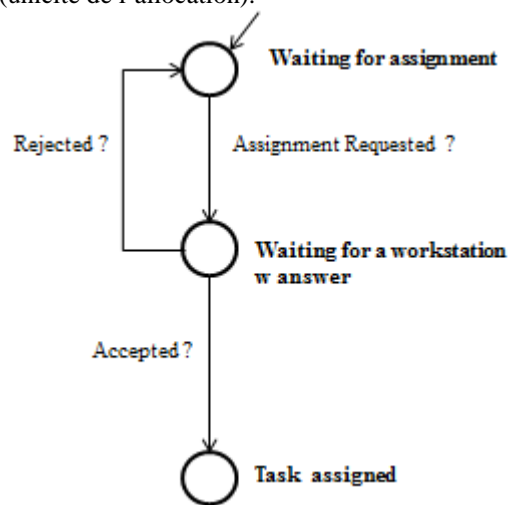


Figure 3 : Modèle générique d'une tâche

#### 4.3.2 Modèle générique d'un poste de travail

Le modèle générique VM d'un poste de travail  $w_k$  est défini par un automate avec deux localités. Il est présenté ci-dessous et sur la figure 4. Le poste va recevoir les demandes des tâches et répondre en fonction des contraintes définies précédemment (C1 et C2).

- Dans la première localité, le poste de travail  $w_k$  est en attente d'une demande d'allocation. Une fois la requête d'une tâche reçue, le modèle WM atteint la localité nommée « Calcul de la réponse ».
- Deux transitions partent de cette transition, mais les gardes, qui sont exclusives l'une par rapport à l'autre, ne permettent d'en franchir qu'une seule en fonction des contraintes :
  - si l'expression logique (C1 et C2) est fautive, le poste de travail émet un message de rejet de la demande et retourne dans sa localité initiale ;
  - si C1 et C2 sont vérifiées simultanément, le poste de travail accepte et envoie un message d'acceptation.

Si l'allocation a été acceptée, la valeur  $A(t_i) = w_k$  est enregistrée.

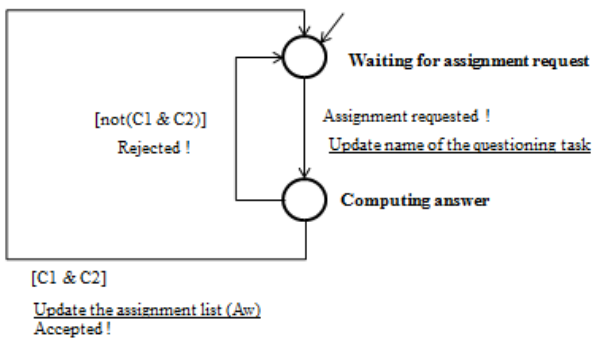


Figure 4 : Modèle générique d'un poste de travail

#### 4.3.3 Initialisation, instanciation, modèle générique de la ligne

Le modèle complet de notre système est composé par un réseau formé d'instances des deux modèles précédents.

$(TM_1 || TM_2 || \dots || TM_{Tmax} || WM_1 || WM_2 || \dots || WM_{Wmax})$ :

- $(Tmax - m)$  instances du modèle de tâches avec  $m$  le nombre de tâches déjà terminées lors du rééquilibrage,
- $Wmax$  instances du modèle de poste de travail (la capacité initiale des postes est adaptée pour tenir compte des tâches déjà terminées).

La synchronisation est assurée par l'envoi et la réception de messages (« demande d'allocation », « rejetée », « acceptée ») De plus, pour éviter qu'une tâche effectuée une demande alors qu'une n'a pas encore reçu de réponse, la localité « calcul de la réponse » est verrouillée : une réponse doit être envoyée pour pouvoir quitter cette localité et pouvoir faire une nouvelle demande.

#### 4.3.4 Obtention d'une solution

Une solution acceptable (respectant toutes les contraintes, mais pas forcément l'optimale) est obtenue s'il existe un état du système où toutes les tâches ont atteint leur localité « tâche allouée ». La valeur des variables enregistrées pendant l'exécution de la trace ayant conduit à cet état forme la solution recherchée.

La vérification de modèle (model-checking) est une technique formelle qui explore l'espace d'état d'un modèle SED (système à événements discrets) pour identifier si la valeur de certaines propriétés sur l'ensemble ou une partie de l'espace d'état. Ces propriétés sont exprimées en utilisant une logique temporelle.

Cette technique peut aisément être utilisée pour de la recherche d'atteignabilité avec une stratégie en profondeur pour éviter l'explosion combinatoire. Pour cela, on écrit la propriété suivante en utilisant CTL expression (Computation Tree Logic) (Clarke et Emerson, 1982) :

(P1)  $EF(\ll \text{Toutes les tâches sont allouées} \gg)$

avec E le quantifieur d'existence d'un chemin et F le quantifieur temporel de l'éventualité. La propriété P1 signifie donc : il existe un chemin tel que « toutes les tâches sont allouées » sera vraie un jour.

## 5 CAS D'ÉTUDE, APPLICATION AU CAS DE LA SOCIÉTÉ TRANE

### 5.1 Présentation de la société Trane

Pour tester notre approche, nous nous sommes appuyé sur un exemple de l'entreprise Trane. Celle-ci conçoit, assemble et installe des systèmes de climatisation et de conditionnement de l'air de toutes dimensions. Elle dispose de plusieurs usines dans le monde, dont deux en France fabriquant les grands modèles de la gamme. Dans ces usines, les produits sont assemblés sur des lignes manufacturières organisées conformément aux concepts de la DFT (Demand Flow Technology). À cause de la grande dimension des produits, aucun stock intermédiaire n'est fait entre deux postes de la ligne et le ré-séquencement des produits déjà commencé n'est pas autorisé. Pour cela, les produits changent de postes simultanément selon une cadence calculée en fonction du temps d'ouverture et du nombre de produits à fabriquer : le takt time.

### 5.2 Description du cas d'étude

Dans notre cas d'étude, nous étudions les trois premiers postes d'une ligne qui en compte quatorze. Ces trois premières opérations regroupent toutes les tâches qui doivent être effectuées obligatoirement avant de tester le produit sur le quatrième poste. La description des tâches est donnée dans la table 1.

Nom	Description	Durée (min)	Poste pour l'assignation initiale
1	Prépa & pose Base	3,3	1
2	Pose Evaporateur	6,4	1
3	Bouteille C1	4,4	1
4	Bouteille C2	4,4	1
5	Brasage ligne liquide C1	5,3	1
6	Brasage ligne liquide C2	5,3	1
7	Pose Ligne SLHX (Aspi - Liquide) C1	3,97	1
8	Pose Ligne SLHX (Aspi - Liquide) C2	3,97	1
9	Brasage ligne SLHX C1	2,78	1
10	Brasage ligne SLHX C2	2,78	2
11	Pose Vanne 4 voies C1	5,4	1
12	Pose Vanne 4 voies C2	6,2	1
13	Brasage ligne Vanne 4 Voies C1	11,7	1
14	Brasage ligne Vanne 4 Voies C2	11,7	3
15	Pose compresseurs C1	22,55	2
16	Pose compresseurs C2	22,55	2
17	Brasage ligne Compresseur C1	16,2	2
18	Brasage ligne Compresseur C2	16,2	3
19	Profilés de Structure Toit	5,2	3
20	Assemblage ligne Entrée d'eau	6,5	3
21	Assemblage ligne Sortie d'eau	10,7	3
22	Isolation ligne Entrée eau évaporateur	9,6	3
23	Isolation ligne Sortie eau évaporateur	5,9	3

Tableau 1 : Description des tâches

Ces tâches doivent respecter les contraintes de précedence présentée sur la figure 5.

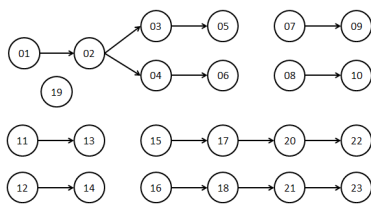


Figure 5 : Graphe de précedence

Le takt time calculé par l'entreprise est de 67 minutes (4020 secondes). L'équilibrage initial fourni par

l'entreprise est représenté sur la figure 6 (écart maximum par rapport au takt de 228 secondes ( $\epsilon$ )).

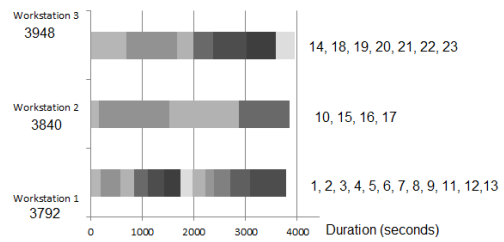


Figure 6. Équilibrage initial de la ligne

### 5.3 Implémentation sur le model-checker UPPAAL

#### 5.3.1 Première étape, l'initialisation

Le modèle générique présenté dans la section 3 est implémenté avec le logiciel UPPAAL (figure 7). Lorsqu'un retard est détecté, on instancie le modèle générique d'une tâche, autant de fois qu'il y a de tâches qui peuvent être réallouées, c'est-à-dire « 23-m » fois avec le nombre de tâches déjà terminées lors du calcul du rééquilibrage. De plus, on crée 3 instances du modèle générique d'un poste de travail en adaptant leur capacité initiale pour prendre en compte la durée des tâches déjà terminées. Des variables de construction sont ajoutées aux éléments mentionnés précédemment comme par exemple un état dit « urgent » qui permet de ne pas passer par une autre transition tant que la question posée n'a pas reçu de réponse ; ou encore une variable contenant l'identifiant de la tâche pour la communication entre les automates ; ou aussi une liste pour visualiser la solution d'allocation.

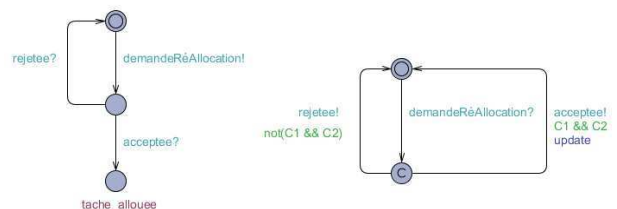


Figure 7 : Modèle générique d'une tâche et d'un poste de travail dans le logiciel UPPAAL

#### 5.3.2 Seconde étape, la résolution

Une fois le modèle initialisé, nous lançons la résolution. La propriété CTL présentée dans la section 4 (P1) s'écrit, dans la syntaxe du logiciel :

$E\langle \rangle$  forall (i:id\_T)Task(i).allouée

La solution est obtenue en demandant au logiciel de conserver une trace (diagnostic trace) en plus de la vérification de la propriété.

## 5.4 Résultats

### 5.4.1 Exemple 1

Dans ce premier exemple, le retard est détecté lors de la réalisation de la tâche 3 qui a lieu sur le premier poste. Les trois premières tâches ne peuvent donc pas être déplacées car elles sont terminées ou en cours de réalisation. De plus la capacité du premier poste de travail est réduite pour tenir compte de ces tâches.

Avec un retard estimé à 60 secondes, notre méthode trouve, en quelques secondes, qu'une nouvelle allocation est possible : il faut déplacer la tâche n°9 en la retirant du premier poste pour la planifier sur le deuxième. Nous obtenons alors le rééquilibrage présenté sur la figure 8.

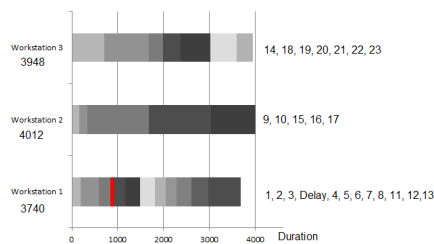


Figure 8 : Rééquilibrage pour un retard de 60 secondes sur la troisième tâche

### 5.4.2 Exemple 2

Pour ce deuxième exemple, le retard est détecté lors de la réalisation de la tâche 15 qui a lieu sur le deuxième poste. Les quinze premières tâches ne peuvent donc pas être déplacées car elles sont terminées ou en cours de réalisation. De plus les capacités des deux premiers postes de travail sont réduites pour tenir compte de ces tâches.

Avec un retard estimé à 120 secondes, le délai est absorbé sans modifier l'équilibrage, c'est pourquoi notre méthode nous donne une réallocation des tâches identique à l'équilibrage initial.

Avec un retard estimé supérieur à 165 secondes, il n'existe pas de solutions acceptable : le takt time sera forcément dépassé et notre méthode ne retourne aucune solution.

## 6 CONCLUSIONS ET PERSPECTIVES

Dans cet article, nous avons montré comment la recherche d'atteignabilité pouvait être utilisée pour le rééquilibrage dynamique d'une ligne d'assemblage. Dans le cadre d'un système manufacturier intelligent, cette méthode permet de donner rapidement une solution acceptable pour adapter localement et de manière réactive l'équilibrage optimal initialement défini.

La méthode permet de trouver une solution si celle-ci existe. Cependant, lorsqu'il n'y a plus assez de temps disponible, le retard ne peut pas être résolu par une

simple réallocation des tâches. C'est pourquoi nos futurs travaux ajouteront la possibilité de paralléliser des tâches.

De plus, comme parfois le déplacement de certaines tâches peut être plus coûteux que de laisser le déséquilibre temporairement (et donc un retard), il faudra tenir compte du coût de reconfiguration afin de proposer à l'entreprise une solution plus proche de l'optimal.

## REFERENCES

- Alur, R., Dill, D.L., 1994. A theory of timed automata. *Theoretical Computer Science* 126, 183–235. doi:10.1016/0304-3975(94)90010-8
- Battaïa, O., Dolgui, A., 2013. A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics* 142, 259–277. doi:10.1016/j.ijpe.2012.10.020
- Becker, C., Scholl, A., 2006. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research* 168, 694–715. doi:10.1016/j.ejor.2004.07.023
- Behrmann, G., Brinksma, E., Hendriks, M., & Mader, A. (2005, April). Production scheduling by reachability analysis—a case study. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International* (pp. 140a-140a). IEEE.
- Boysen, N., Fliedner, M., Scholl, A., 2007. A classification of assembly line balancing problems. *European Journal of Operational Research* 183, 674–693. doi:10.1016/j.ejor.2006.10.010
- Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: Which model to use when? *International Journal of Production Economics* 111, 509–528. doi:10.1016/j.ijpe.2007.02.026
- Clarke, E.M., Emerson, E.A., 1982. Design and synthesis of synchronization skeletons using branching time temporal logic, in: Kozen, D. (Ed.), *Logics of Programs, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 52–71
- Estellon, B., Gardi, F., Nouioua, K., 2008. Two local search approaches for solving real-life car sequencing problems. *European Journal of Operational Research* 191, 928–944. doi:10.1016/j.ejor.2007.04.043
- Falkenauer, E., 2005. Line balancing in the real world. Presented at the PLM'05: international conference on product life cycle management, pp. 360–370.



- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5, 287-326.
- Grangeon, N., Leclaire, P., Norre, S., 2011. Heuristics for the re-balancing of a vehicle assembly line. *International Journal of Production Research* 49, 6609–6628. doi:10.1080/00207543.2010.539025
- Lematre, T., 2013. Allocation de fonctions de commande de systèmes critiques par recherche d'atteignabilité dans un réseau d'automates communicants. École normale supérieure de Cachan - ENS Cachan.
- Makssoud, F., Battaïa, O., Dolgui, A., 2013. Reconfiguration of Machining Transfer Lines, in: Borangiu, T., Thomas, A., Trentesaux, D. (Eds.), *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics, Studies in Computational Intelligence*. Springer Berlin Heidelberg, pp. 339–353.
- Marangé, P., Pétin, J.-F., Manceaux, A., Gouyon, D., 2011. Contribution à la reconfiguration des systèmes de production. *Journal Européen des Systèmes Automatisés* 45, 45–60. doi:10.3166/jesa.45.45-60
- Scholl, A., 1999. Balancing and sequencing of assembly lines (Publications of Darmstadt Technical University, Institute for Business Studies (BWL)). Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).
- Scholl, A., Becker, C., 2006. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research* 168, 666–693. doi:10.1016/j.ejor.2004.07.022
- Subbiah, S., Engell, S., 2010. Short-Term Scheduling of Multi-Product Batch Plants with Sequence-Dependent Changeovers Using Timed Automata Models, in: S. Pierucci and G. Buzzi Ferraris (Ed.), *Computer Aided Chemical Engineering*. Elsevier, pp. 1201–1206.
- Vain, J., Randvee, I., Riismaa, T., Ernits, J., 2002. Solving line balancing problems with model checking., in: *Proceedings of the Estonian Academy of Sciences, Engineering*.