

# Problème d'acquisition de données par une torpille

G. Simonin, A.-E. Baert, A. Jean-Marie and R. Giroudeau

LIRMM, CNRS, Université Montpellier 2  
161, rue Ada, 34392 Montpellier Cedex 5 - France.  
Auteur à contacter : [gilles.simonin@lirmm.fr](mailto:gilles.simonin@lirmm.fr)

## 1 Introduction

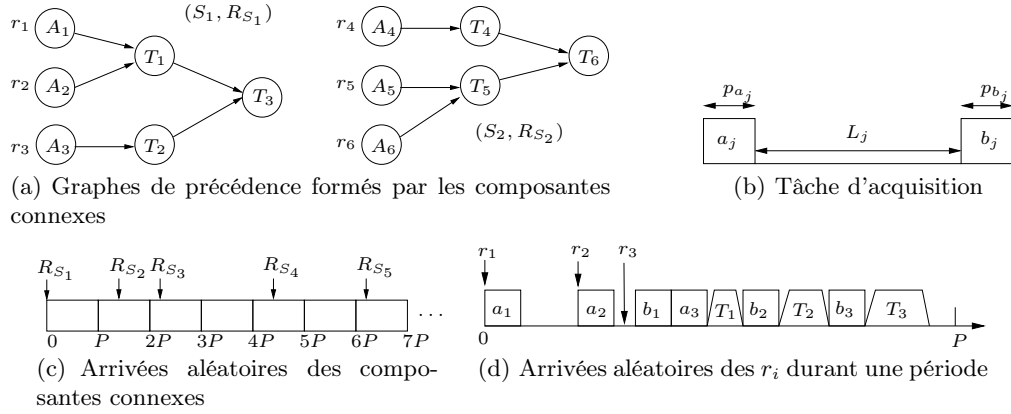
Ce papier traite du problème d'ordonnancement stochastique de tâche-couplées, introduites par Shapiro [1], qui sont soumises à des contraintes de précédence avec des tâches dites de traitement sur un monoprocesseur. Le problème que nous proposons ici est celui de l'acquisition de données par une torpille autonome sous-marine. Cette torpille a comme objectif d'effectuer différents relevés, d'analyser la topologie du fond sous-marin, ou encore faire de la cartographie. Une fois la torpille en immersion, elle va recevoir de manière aléatoire différents groupes de tâches à réaliser dans un certain ordre sur son monoprocesseur. Ces tâches vont devoir être réalisées de manière périodique jusqu'à la fin de l'expérience. Afin d'acquérir les données demandées, puis de pouvoir les traiter, la torpille possède des capteurs qui permettent soit d'envoyer des échos qui reviennent après un certain temps d'inactivité, soit de faire des relevés en moyenne (moyenne entre deux relevés, ce qui revient à une tâche-couplée). Le problème, qui sera noté  $\mathcal{TP}$ , consiste à décider si, pour certains groupes de tâches à réaliser, la torpille peut exécuter périodiquement toutes les tâches dans le temps imparti, et ce jusqu'à l'arrivée des dernières tâches à effectuer. La problématique définie ci-dessus étant typiquement un problème d'ordonnancement stochastique sur un monoprocesseur, nous allons modéliser ce problème puis conclure en présentant de manière synthétique l'approche utilisée pour le résoudre.

## 2 Modélisation du problème

La modélisation du problème d'acquisition de données par une torpille en immersion  $\mathcal{TP}$  est la suivante.

Soit  $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$  l'ensemble des composantes connexes, chaque composante connexe  $S_i$  ( $1 \leq i \leq m$ ) forme un graphe de précédence orienté entre certaines tâches d'acquisition et certaines tâches de traitement (voir Figure 1(a)). Dans un premier temps et pour simplifier cette première étude, chaque  $S_i$  possède une date périodique  $P_i = P$  à la fin de laquelle toutes les tâches d'acquisition doivent être re-exécutées. Enfin, les composantes connexes possèdent chacune une date d'arrivée dans le système de manière aléatoire en suivant une loi uniforme (voir Figure 1(c)).

Soit  $\mathcal{A} = \{A_1, A_2, \dots, A_{n_1}\}$  l'ensemble des tâches d'acquisition, semblables aux tâches-couplées introduites par Shapiro [1]. Chaque tâche d'acquisition  $A_j$  ( $1 \leq j \leq n_1$ ) consiste en deux sous-tâches  $a_j$  et  $b_j$  de durée d'exécution respectives  $p_{a_j}$  et  $p_{b_j}$ . Elles sont séparées par un délai d'inactivité incompressible  $L_j$ , i.e., la sous-tâche  $b_j$  doit démarrer son exécution

FIG. 1. Modélisation du problème  $\mathcal{TP}$ 

exactement  $L_j$  unités de temps après l'exécution de la sous-tâche  $a_j$  (voir Figure 1(b)). Les tâches d'acquisition seront toujours des sources dans le graphe de précedence des composantes connexes. Enfin, chaque tâche d'acquisition  $A_j$  possède une date d'arrivée notée  $r_{A_j}$  constante à chaque période et attribuée aléatoirement selon une loi uniforme (voir Figure 1(d)).

Soit  $\mathcal{T} = \{T_1, T_2, \dots, T_{n_2}\}$  l'ensemble des tâches de traitement. Ces tâches ont la propriété d'être préemptibles et sont toujours précédées par au moins une tâche dans les graphes de précedence. Chaque tâche de traitement  $T_k$  ( $1 \leq k \leq n_2$ ) possède une longueur d'exécution variable notée  $p_{T_k}$ .

### 3 Résultats

Dans ce résumé, nous présentons un nouveau type de problème d'ordonnement en temps réel. Dans [3], nous avons montré que le même problème déterministe était  $\mathcal{NP}$ -complet. Afin de décider de la faisabilité du problème, nous cherchons différents ordonnancements utilisant des règles de priorité proche de celle utilisée dans [2]. L'ordonnement le plus intéressant est d'utiliser les graphes de précedence disponibles par la torpille entre deux périodes. En effet, à chaque période, nous appliquons une règle de priorité en fonction des arrivées aléatoires  $r_j$  des tâches d'acquisition  $A_j$  dans les graphes de précedence, qui permet de calculer l'ordre d'exécution des tâches contenues dans les composantes connexes déjà arrivées.

### Références

1. R.D. Shapiro : Scheduling coupled tasks, Naval Research Logistics Quarterly, 27, 1980, 477-481.
2. M. Pinedo : Scheduling, Theory, Algorithms, and Systems, Second Edition, Prentice Hall, 2006.
3. G. Simonin : Étude de la complexité de problèmes d'ordonnement avec tâches-couplées sur monoprocesseur, Majestic '08, 2008.