

---

# Planification stratégique du système robotique

Ce chapitre aborde les décisions stratégiques définies dans l'architecture de décisions du chapitre 3, en particulier la gestion d'autonomie des batteries, la configuration des robots et la localisation des robots. Ces décisions sont prises sur un horizon d'un ou plusieurs postes de travail (shifts) afin de prendre en compte la variation des activités dans le temps. Ce chapitre commence par une définition détaillée des problèmes relatifs à ces trois grandes décisions. Nous proposons ensuite trois approches fondées sur des modèles de programmation mathématique. Ces trois approches sont ensuite comparées à travers différentes instances numériques.

## **4.1. Les trois sous problèmes de la planification stratégique**

### **4.1.1. Gestion d'autonomie des batteries**

Un des problèmes majeurs identifiés durant la première réunion d'évaluation du projet est la durée relativement conséquente de rechargement des batteries, et la situation, dans laquelle tous les robots se vident de leur énergie et se rechargent ensembles durant la même période, est à éviter. Cette situation devrait être évitée afin de garder des robots opérationnels pour assurer la continuité d'assistance au personnel en cas de missions urgentes.

Plusieurs solutions ont été proposées afin de remédier à ce problème d'autonomie, certaines par le groupe hardware, en proposant d'augmenter les capacités des robots en ajoutant des batteries supplémentaires afin de prolonger la durée d'activité des robots ou bien d'installer des stations de rechargement automatique qui permettent aux robots de se recharger sans intervention humaine durant les périodes creuses où les robots sont en veille en attendant l'arrivée de nouvelles missions.

Notre groupe « swarm intelligence » s'est également penché sur ce problème et a proposé de planifier les périodes de rechargement de chaque robot afin d'assurer un service minimum nécessaire tout au long de la journée.

L'agent de décision consacré à la gestion de l'autonomie des robots, doit gérer et planifier les cycles de rechargement, et s'assurer de la disponibilité d'un nombre suffisant de robots opérationnels durant les futures périodes pour exécuter les futures missions.

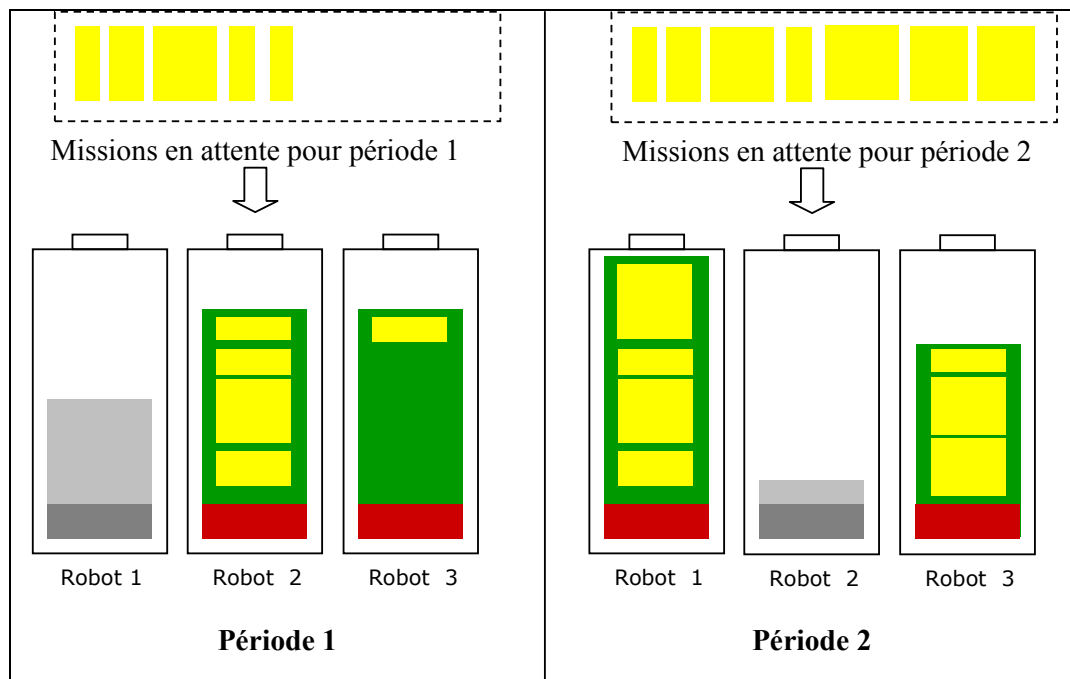


Figure 11 : Gestion de l'autonomie des robots

Les caractéristiques de cet agent de décision sont :

**Évènement déclencheur** : Le processus de planification est déclenché lorsqu'une différence majeure est constatée entre les demandes prévues et les demandes observées qui rendent les plans calculés obsolètes, ainsi que le début d'une tranche horaire où une nouvelle équipe de travail prend le relais.

**Nature du processus de prise de décision** : Comme la gestion d'autonomie doit prendre en compte l'état de l'ensemble des robots et la prévision de l'ensemble des demandes, un processus de décision centralisé semble être le plus adapté pour une telle situation.

**Horizon de planification** : compte tenu des temps de chargement importants (plus de 8h) et de l'objectif d'assurer la disponibilité des robots, l'horizon de planification doit s'étendre sur plusieurs périodes consécutives.

**Données nécessaires** : Les paramètres qui doivent être pris en compte pour le calcul des plans de chargement sont : le nombre de robots, la capacité de stockage d'énergie des batteries embarquées sur les robots, le niveau d'énergie de chaque robot, la vitesse de rechargement de chaque robot, la vitesse moyenne de déchargement des robots en mode opérationnel et en mode veille, les prévisions de demande et l'énergie nécessaire pour accomplir ces demandes. La configuration des robots et les modules attachés aux robots peuvent être pris en considération pour le calcul de l'affectation des tâches si ces informations peuvent être disponibles à ce stade de calcul.

**Décisions** : A la fin du calcul, l'agent de décision doit établir, pour chaque période prise en considération, un plan de rechargement spécifiant les robots destinés à se recharger, ainsi que la charge affectée à chaque robot opérationnel.

### 4.1.2. Configurations des robots

Une des innovations du projet IWARD est l'ensemble des fonctionnalités/ modules « *plug and play* » (brancher et faire fonctionner) qui ressemblent à des tiroirs et qui peuvent être ajoutés/ enlevés à chaud par n'importe quel personnel disponible. Si par exemple il a été décidé de reconfigurer un robot en lui ajoutant la fonctionnalité de transport de médicament, il suffit qu'un(e) infirmier(e) prenne le tiroir/module qui correspond à cette fonctionnalité et le glisse dans l'un des endroits prévus. Comme les demandes d'assistance varient dans une journée et que certaines demandes sont périodiques, les robots doivent être reconfigurés différemment durant la journée. La configuration d'un robot nécessite une intervention humaine et doit donc être planifiée de préférence à des moments précis comme lors des changements de postes. Puisque la configuration d'un robot détermine sa charge, il est préférable de planifier les configurations des robots en fonction des prévisions de demandes sur plusieurs périodes. Ce problème peut aussi être couplé avec le problème de l'autonomie des robots, présenté précédemment, afin que la capacité énergétique des robots et le type des missions affectées soient pris en considération lors de la configuration ainsi que l'affectation des missions.

Le but de l'agent de décision, responsable de la reconfiguration, est d'analyser la configuration actuelle des robots, et de les reconfigurer en fonction des demandes futures et de la disponibilité des fonctionnalités.

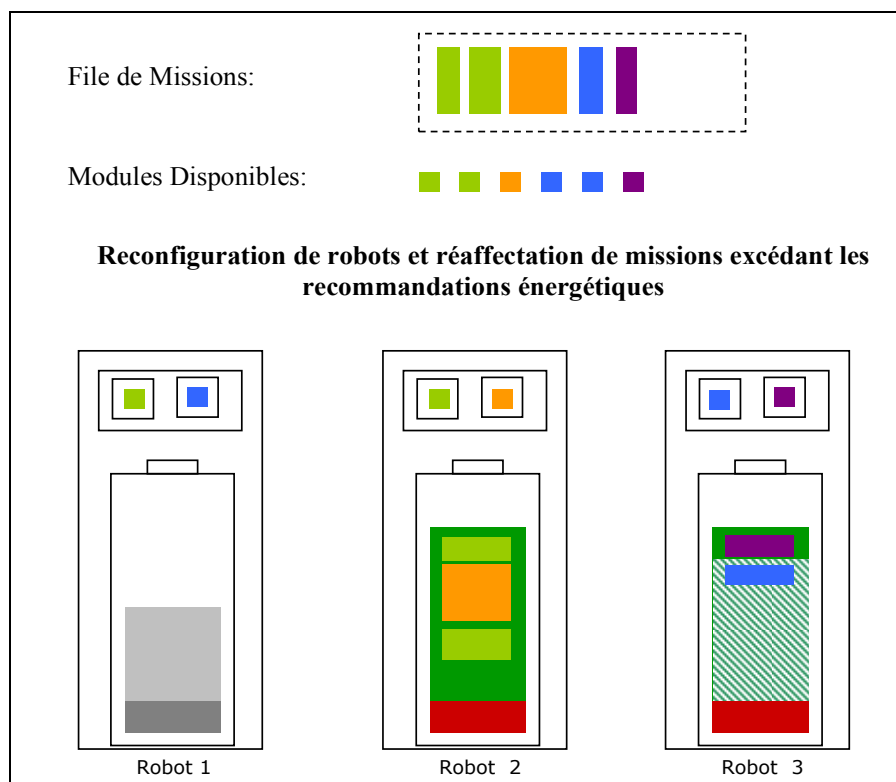


Figure 12 : Configuration et autonomie des robots.

Les caractéristiques de cet agent de décision sont les suivantes :

**Événement déclencheur :** Le processus de reconfiguration est déclenché soit au commencement d'une nouvelle période, ce qui indique le changement des équipes de travail ainsi que le changement des demandes du personnel ; soit un changement de ressources

disponibles qui est généralement un changement de robots opérationnels suite à un changement des plans de rechargement des robots.

**Horizon de planification** : L'horizon de planification peut être limité à une seule période ou étendu à plusieurs périodes. Le processus de reconfiguration sera déclenché à la fin de chaque période.

**Nature du processus de prise de décision** : Sachant que ce processus doit tenir compte de l'état de tous les robots du système, et que les robots doivent être configurés afin d'assurer une disponibilité globale par rapport à la prévision de demandes, un processus de type centralisé semble plus pertinent pour effectuer cette tâche.

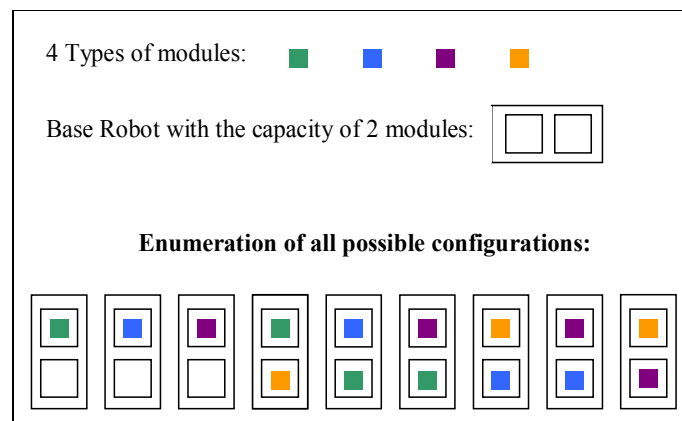


Figure 13 : Enumération des configurations possibles d'un robot ayant 2 tiroirs disponibles

**Données nécessaires** : Les données essentielles pour effectuer le calcul de reconfiguration sont : le nombre de robots opérationnels ainsi que la capacité énergétique de chaque robot, le nombre des fonctionnalités disponibles ainsi que leur type, l'ensemble de configurations possibles pour chaque robot, les missions prévues, ainsi que leur type et la consommation énergétique nécessaire pour exécuter ces missions.

**Décisions** : Les décisions prises par cet agent concernent la configuration de chaque robot et les modules qui lui seront affectés.

### 4.1.3. Localisation des stations des robots

Le positionnement des robots dans l'hôpital influence la réactivité du système robotique. Des stations d'attente doivent être positionnées de manière intelligente dans l'hôpital, ainsi les robots y seront affectés lorsqu'ils sont en état de veille, dans le but de mieux répondre aux demandes urgentes et de minimiser les parcours dans le vide.

Lorsqu'un robot termine toutes les missions qui lui ont été affectées, il repart à sa station d'attente. Cette station doit être choisie en fonction de la proximité des futures missions prévues qui lui seront affectées.

Ces stations d'attente évitent que les espaces vitaux de l'hôpital ne soient encombrés par des robots en état de veille. Ces stations peuvent aussi jouer un rôle de station de rechargement si elles sont équipées de « self docking station » ou station de chargement automatique qui permet aux robots de se ressourcer en énergie sans intervention humaine.

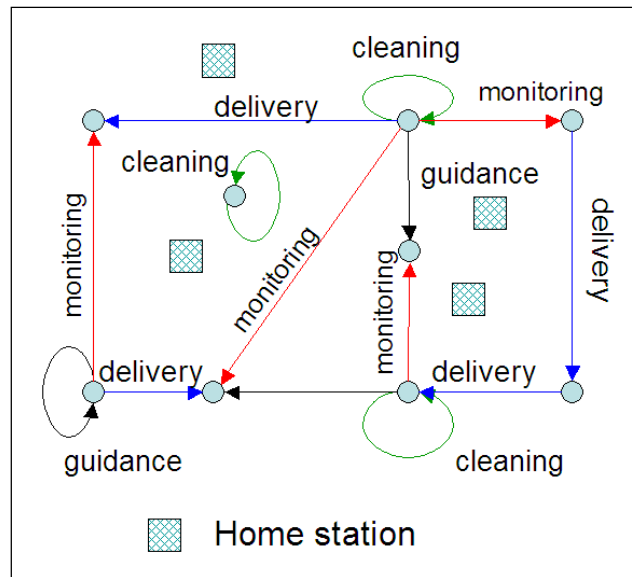


Figure 14: Localisation des stations d'attente en fonction de futures demandes

Les caractéristiques de l'agent de décisions responsable de la localisation des ces stations d'attente sont :

**Événement déclencheur** : Le processus de localisation des stations est déclenché, suite à un changement des ressources existantes (dans notre cas, suite à un changement des robots opérationnels qui peut être engendré par une modification des plans de rechargement des robots). Un changement des fonctionnalités existantes sur les robots peut aussi nécessiter un changement des stations d'attente, par exemple le robot équipé avec le module de transport peut être mieux placé à côté de la pharmacie qui est le premier demandeur de ce module. Également, le processus de localisation peut être déclenché suite à un changement des demandes du personnel, souvent lié à un changement d'équipe durant la journée, ou la passation entre deux équipes qui marque le début d'une nouvelle période et donc nécessite une ré-optimisation du système.

**Horizon de planification** : Cet agent effectue ses calculs sur une période, en tenant compte des capacités énergétiques des robots.

**Nature de l'agent de décision** : Vu la nécessité de prendre en compte l'état de l'ensemble des robots et la nécessité de couvrir de manière optimale les demandes, un agent de décision de type centralisé, peut être plus capable d'effectuer l'association entre les robots et les stations d'attente les plus appropriées.

**Données nécessaires** : l'ensemble des lieux qui peuvent servir de stations d'attente, les futures demandes prévues ainsi que leur nature, les points de départ et d'arrivée de ces missions, la fréquence ou la probabilité d'occurrence de ces missions, les temps d'exécution de ces missions, la matrice du temps nécessaire pour se déplacer d'un point d'intérêt à un autre.

Ce problème peut être couplé avec le problème de configuration et même avec le problème d'autonomie des robots afin de s'assurer que les missions seront affectées à un robot opérationnel proche ayant assez d'énergie stocké pour exécuter sa mission

**Décision** : Affecter à chaque robot opérationnel une station d'attente vers laquelle il doit se diriger lorsqu'il est en état de veille.

## 4.2. Planification stratégique intégrée / combinée

L'un des avantages du regroupement des trois problèmes est de pouvoir optimiser de manière intégrée l'ensemble des décisions stratégiques en tenant compte des contraintes d'énergie, de configuration et de localisation. L'inconvénient majeur est la taille importante du problème.

Nous considérons un système robotique composé d'un ensemble  $N$  de robots de base, qui sont dotés de batteries ayant une capacité énergétique limitée et dont la durée de rechargement est assez importante. Chaque robot de base peut être configuré différemment en y ajoutant des fonctionnalités. Soit  $F$  l'ensemble des types de ces fonctionnalités et  $N_f$  le nombre d'exemplaires disponibles de modules du type  $f$ .

Désignons par  $C$  l'ensemble de configurations possibles pour un robot de base. Cet ensemble peut être déterminé par énumération des différentes combinaisons de fonctionnalités par rapport au nombre d'emplacements possibles sur chaque robot ainsi que la compatibilité des fonctionnalités. Chaque configuration  $c \in C$  est définie par un vecteur d'entiers  $[a_{cf}]$  indiquant le nombre d'exemplaires de chaque fonctionnalité qui constitue la configuration  $c$ .

Soit  $H$  l'ensemble des stations d'attente où les robots peuvent attendre l'arrivée de nouvelles missions. En recevant une nouvelle mission, le robot devient opérationnel, quitte la station en se dirigeant vers le point de départ de la mission puis exécute la mission. S'il n'a pas d'autres missions affectées, il retourne à sa station d'attente  $h$ .

On considère que chaque journée est divisée en plusieurs créneaux  $t \in T$ . Chaque créneau a une longue  $T_{max}$  et un nombre minimal  $L_t$  de robots opérationnels pour assurer un service minimal durant la période  $t$ .

Chaque robot  $n$  est doté d'une batterie. Soit  $Q_{n,0}$  le niveau d'énergie initial du robot  $n$ , Soit  $l_n$ , la quantité d'énergie consommée par un robot  $n$  en état de veille durant un créneau  $t$  et soit  $G_n$  la quantité d'énergie gagnée par le rechargement d'un robot  $n$  dans un créneau.

On désigne par  $e_{n,m}$  la quantité d'énergie dépensée par un robot  $n$  pour exécuter une mission  $m$ . Plus précisément, chaque robot  $n$  consomme au moins une quantité  $l_n$  d'énergie et consomme en plus une quantité  $e_{n,m}$  d'énergie additionnelle pour chaque mission exécutée.

Chaque mission  $m$  est caractérisée par les paramètres suivants :

- $\square_{m,t}$ : fréquence d'occurrence de la mission  $m$  durant la période  $t$ ,
- $T_m$ : durée nécessaire pour exécuter la mission  $m$ ,
- $C_m$ : ensemble des configurations permettant l'exécution de la mission  $m$ ,
- $A_{hm}$ : temps nécessaire pour un robot afin d'effectuer le trajet de la station d'attente  $h$  jusqu'au point de départ de la mission  $m$ ,
- $R_{hm}$ : temps nécessaire à un robot pour effectuer le trajet du retour du point de terminaison de la mission  $m$  jusqu'à la station d'attente  $h$

Le temps total d'exécution d'une mission est ainsi décomposé en trois temps : temps de réaction  $A_{hm}$ , durée d'exécution  $T_m$  et temps de retours  $R_{hm}$ . Si la mission  $m$  est affectée au robot  $n$ , la station d'attente qu'il faut considérer pour les calculs de ces durées est la station  $h$  à laquelle le robot  $n$  est assigné. Ces durées sont calculées en fonction de la vitesse moyenne des robots et de la longueur du trajet.

On désigne par  $Q_{n,max}$  le niveau maximal d'énergie que la batterie du robot  $n$  peut accumuler, et  $Q_{n,min}$  le niveau minimal d'énergie à ne pas atteindre afin de garder une quantité suffisante d'énergie pour faire face à certains imprévus et pouvoir rentrer se recharger avant que sa batterie ne soit asséchée complètement.

Deux types de charges peuvent être considérés : la charge de travail  $\phi_m(A_{hm}+T_m+R_{hm})$  induit par l'exécution d'une mission  $m$  et la charge énergétique  $\phi_m e_{nm}$ . La somme des charges de travail d'un robot pendant une période doit être inférieure à  $T_{max}$  qui est proportionnelle à la durée de cette période. La quantité d'énergie restante due à la somme des charge énergétique des missions affecté au robot ne doit pas atteindre le seuil minimal  $Q_{n,min}$  de la batterie.

#### 4.2.1. **Modèle mathématique de la planification stratégique intégrée**

Les décisions tactiques/opérationnelles comme l'ordonnancement des missions ne sont pas considérés à ce niveau. Les coûts de reconfiguration et de relocalisation des stations d'attente sont supposés nuls. Le rechargement partiel est interdit afin de maximiser la durée de vie des batteries des robots

Le problème présenté consiste à déterminer, pour chaque période, les robots à recharger, et pour les robots opérationnels, une configuration de fonctionnalités attachées au robot ainsi qu'une station d'attente.

Le but ultime est d'exécuter toutes les missions au moindre coût, tout en respectant les contraintes physiques comme le nombre maximal  $H_h$  de robots qu'une station  $h$  peut accueillir, la disponibilité des fonctionnalités.

Afin de formuler le problème sous forme d'un programme linéaire en nombre entier les variables de décision suivantes sont utilisées:

$$x_{nht} = \begin{cases} 1, & \text{si le robot } n \text{ est stationné en } h \text{ durant la periode } t \\ 0, & \text{sinon.} \end{cases}$$

$$y_{nct} = \begin{cases} 1, & \text{si le robot } n \text{ utilise la configuration } c \text{ durant la periode } t \\ 0, & \text{sinon.} \end{cases}$$

$$z_{nmt} = \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot } n \text{ durant la periode } t \\ 0, & \text{sinon.} \end{cases}$$

$$w_{nt} = \begin{cases} 1, & \text{si le robot } n \text{ se recharge durant la periode } t \\ 0, & \text{si le robot } n \text{ est opérationnel durant la periode } t. \end{cases}$$

$$u_{mt} = \begin{cases} 0, & \text{si la mission } m \text{ est affectée à un robot durant la période } t. \\ 1, & \text{sinon.} \end{cases}$$

On utilise aussi la quantité  $q_{n,t}$  d'énergie à la fin de la période  $t$  et la quantité  $g_{nt}$  d'énergie acquise par le robot  $n$  en se rechargeant durant la période  $t$ .

$$\text{Minimize } \sum_{t \in T} \sum_{n \in N} \sum_{h \in H} \sum_{m \in M} \phi_{m,t} (\alpha A_{hm} + T_m + \beta R_{hm}) r_{nhm,t} + \sum_{t \in T} \sum_{m \in M} p_m \phi_{mt} u_{m,t} \quad (4-1)$$

sous contraintes:

$$\sum_{h \in H} x_{nh,t} + w_{n,t} = 1, \quad \forall n \in N; \forall t \in T \quad (4-2)$$

$$\sum_{c \in C} y_{nc,t} + w_{n,t} = 1, \quad \forall n \in N; \forall t \in T \quad (4-3)$$

$$u_{m,t} + \sum_{n \in N} z_{nm,t} = 1, \quad \forall t \in T; \forall m \in M \quad (4-4)$$

$$\sum_{c \in C} \sum_{n \in N} a_{cf} y_{nc,t} \leq N_f \quad \forall f \in F; \forall t \in T \quad (4-5)$$

$$\sum_{n \in N} x_{nh,t} \leq H_h, \quad \forall h \in H; \forall t \in T \quad (4-6)$$

$$\sum_{h \in H} \sum_{m \in M} \phi_{m,t} (A_{hm} + T_m + R_{hm}) r_{nhm,t} \leq T_{\max} \quad \forall n \in N; \forall t \in T \quad (4-7)$$

$$z_{nm,t} \leq \sum_{c \in C_m} y_{nct} \quad \forall m \in M, \forall n \in N, \forall t \in T \quad (4-8)$$

$$r_{nhm,t} \geq x_{nh,t} + z_{nm,t} - 1 \quad \forall n \in N; \forall h \in H; \forall m \in M; \forall t \in T \quad (4-9)$$

$$\sum_n w_{n,t} \leq N - L_t \quad \forall t \in T \quad (4-10)$$

$$Q_{n,t} = Q_{n,t-1} + g_{n,t} - l_n(1 - w_{n,t}) - \sum_{m \in M_t} e_m z_{n,m} \quad \forall n = 1 \dots N; \forall t \in T \quad (4-11)$$

$$\underline{Q}_{n,\min} \leq Q_{n,t} \leq \bar{Q}_{n,\max} \quad \forall n \in N; \forall t \in T \quad (4-12)$$

$$g_{n,t} \leq G_n w_{n,t} \quad \forall n \in N; \forall t \in T \quad (4-13)$$

$$x_{nh,t}, y_{nc,t}, z_{nm,t}, w_{n,t}, u_{m,t}, r_{nhm,t} \in \{0, 1\} \quad (4-14)$$

$$\forall b \in B, \forall m \in M, \forall n \in N, \forall t \in T$$

$$g_{n,t} \geq 0; Q_{n,t} \geq 0 \quad \forall n \in N; \forall t \in T \quad (4-15)$$

La fonction objectif (4.1) a pour but de minimiser la charge de travail de l'ensemble des robots en prenant en compte le temps de réaction et le temps nécessaire pour le retour et de minimiser le nombre de missions rejetées ce qui est équivalent à maximiser le nombre de missions affectées aux robots. Les facteurs  $\alpha$  et  $\beta$  permettent d'attribuer des priorités aux temps de réaction et de retour et le facteur  $p_m$  permet de tenir compte de l'importance globale des missions. Les contraintes (4.2) vérifient que chaque robot opérationnel est affecté à une station d'attente, les contraintes (4.3) vérifient que chaque robot opérationnel est configuré selon une des configurations valides de l'ensemble  $C$  des configurations. Les contraintes (4.4) assurent que toute mission est soit affectée à un robot soit rejetée. Les contraintes (4.5)



concernent la disponibilité des fonctionnalités, et que le nombre des fonctionnalités utilisées pour établir les différentes configurations des robots ne dépasse pas la quantité disponible. Les contraintes (4.6) garantissent que le nombre de robots affectés à une station d'attente ne dépasse pas les limites prédéfinies. Les contraintes (4.7) garantissent que la charge maximale de travail d'un robot n'est pas dépassée. Les contraintes (4.8) certifient qu'une mission n'est affectée qu'au robot convenablement configuré.

Les contraintes (4.9) assurent la linéarisation du produit des deux variables de décision  $z_{nm,t}$  et  $x_{nh,t}$  en utilisant la variable  $r_{nhm,t}$ . Cette variable de décision  $r_{nhm,t}$  est égale à 1 si et seulement durant la période  $t$  la mission  $m$  est affectée au robot  $n$  et que la station d'attente  $h$  est désignée comme la station du robot  $n$  la. Les contraintes (4.10) vérifient qu'un nombre minimal  $L_t$  de robots est disponible dans chaque période  $t$ . Les contraintes (4.11) impliquent que le niveau d'énergie à la fin d'une période peut être calculé à partir du niveau d'énergie de la période précédente, de la quantité d'énergie acquise en se rechargeant et de la quantité dépensée soit en état de veille, soit en exécutant les missions affectées au robot durant la période en considération. Les contraintes (4.12) vérifient que le niveau d'énergie emmagasiné sur les robots ne dépasse pas les bornes inférieures et supérieures recommandées. Les contraintes (4.13) garantissent que les robots opérationnels ne peuvent pas être rechargés partiellement et que la quantité d'énergie acquise par un robot en état de rechargement ne dépasse pas la vitesse de rechargement du chargeur. Les contraintes (4.14) représentent les contraintes d'intégrité des variables de décisions.

La résolution de ce problème en nombres entiers utilisant des solveurs commerciaux de type « Branch and Bound » notamment CPLEX, n'est possible que pour des problèmes de petite taille. Pour cela, nous proposons une méthode de génération de colonne dans le paragraphe suivant.

#### 4.2.2. Génération de colonnes pour la planification intégrée

Les techniques de génération de colonnes ont été introduites par Dantzig et Wolf [Dantzig 60] pour résoudre les programmes linéaires structurés de grande taille. Gilmore et Gomory ont été les premiers à appliquer ces méthodes pour résoudre le problème de cutting stock [Gilmore 61] ; une synthèse des avancées récentes des techniques de génération de colonnes et de leurs applications peuvent être trouvées dans [Barnhart 98] et [Lubbecke 05].

Afin d'appliquer les méthodes de génération de colonne, une reformulation du problème (1 - 15), permettant de définir le concept de colonne, sera présentée dans cette partie. Dans le cadre de cette étude, une colonne présente l'état d'un robot  $n$  durant une période  $t$  et spécifie la station d'attente  $h$  dans laquelle il sera stationné, sa configuration  $c$  et les différentes fonctionnalités qui seront montées sur ce robot, ainsi que l'ensemble des missions qui seront affectées à ce robot.

Chaque colonne  $p$  est définie par les variables suivantes :

$$X_h^p = \begin{cases} 1, & \text{si la station } h \text{ est désignée comme la station du robot} \\ 0, & \text{sinon.} \end{cases}$$

$$Y_c^p = \begin{cases} 1, & \text{si le robot est configuré selon la configuration } c \\ 0, & \text{sinon.} \end{cases}$$

$$Z_m^p = \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot pour l'exécuter.} \\ 0, & \text{sinon.} \end{cases}$$

$$W^p = \begin{cases} 1, & \text{si le robot est en état de rechargement.} \\ 0, & \text{sinon.} \end{cases}$$

Evidemment, une colonne est faisable si elle est reliée à une seule zone de stationnement, ayant une configuration unique, et que toutes les missions affectées au robot peuvent être effectuées par cette configuration et que les charges de travail et énergétiques ne sont pas dépassées.

Pour une colonne donnée  $p$ , la charge/coût peut être définie par l'équation suivante:

$$b^p = \sum_{h \in H} \sum_{m \in M} \phi_m (\alpha A_{mh} + T_m + \beta R_{mh}) Z_m^p X_h^p \quad (4-16)$$

La nouvelle variable de décision est définie :

$$\lambda^p = \begin{cases} 1, & \text{si la colonne } p \text{ est sélectionnée} \\ 0, & \text{sinon.} \end{cases}$$

Ainsi le problème combiné peut être reformulé en un « **problème maître** » :

$$\text{Min} \sum_{t \in T} \sum_{n \in N} \sum_{p \in \Omega_{nt}} b^p \lambda^p + \sum_{t \in T} \sum_{m \in M} p_m \phi_{mt} u_{mt} \quad (4-17)$$

où  $\Omega_{nt}$  est l'ensemble des colonnes faisables du robot  $n$  pour la période  $t$  et sous contraintes:

$$\sum_{p \in \Omega_{nt}} \lambda^p = 1, \quad \forall n \in N, t \in T \quad (4-18)$$

$$\sum_{n \in N} \sum_{p \in \Omega_{n,t}} Z_m^p \lambda^p + u_{mt} = 1, \quad \forall t \in T, m \in M \quad (4-19)$$

$$\sum_{n \in N} \sum_{p \in \Omega_{nt}} \left( \sum_{c \in C} a_{cf} Y_c^p \right) \lambda^p \leq N_f, \quad \forall f \in F, t \in T \quad (4-20)$$

$$\sum_{n \in N} \sum_{p \in \Omega_{nt}} X_h^p \lambda^p \leq H_h, \quad \forall h \in H, t \in T \quad (4-21)$$

$$Q_{n,t} - Q_{n,t-1} - g_{n,t} + \sum_{p \in \Omega_{nt}} l_n (1 - W^p) \lambda^p + \sum_{p \in \Omega_{nt}} \left( \sum_{m \in M} \phi_{mt} e_m Z_m^p \right) \lambda^p = 0, \quad \forall n \in N, t \in T \quad (4-22)$$

$$g_{nt} \leq \sum_{p \in \Omega_{nt}} G_n W^p \lambda^p, \quad \forall n \in N, t \in T \quad (4-23)$$

$$\underline{Q}_n \leq Q_{n,t} \leq \bar{Q}_n, \quad \forall n \in N, t \in T \quad (4-24)$$

$$\sum_{n \in N} \sum_{p \in \Omega_{nt}} W^p \lambda^p \leq N - L_t, \quad \forall t \in T \quad (4-25)$$

$$\lambda^p \in \{0,1\}, \forall p \in \Omega; u_{mt} \in \{0,1\}, \forall m \in M, t \in T \quad (4-26)$$

$$g_{n,t} \geq 0; Q_{n,t} \geq 0, \quad \forall n \in N; \forall t \in T \quad (4-27)$$

De même que dans la formulation précédente, la fonction (4.17) a pour but de minimiser la charge de travail et de maximiser le nombre de missions affectées aux robots. Les contraintes (4.18) vérifient que quelque soit la période et pour n'importe quel robot un état et un seul doit être attribué au robot. Les contraintes (4.19) assurent que toute mission est soit affectée à un robot soit rejetée. Les contraintes (4.20) concernent la disponibilité des fonctionnalités, et assurent que le nombre des fonctionnalités utilisées pour établir les différentes configurations des robots ne dépasse pas la quantité disponible. Les contraintes (4.21) garantissent que le nombre de robots affectés à une station d'attente ne dépasse pas les limites prédéfinies. Les contraintes (4.22) déterminent le niveau d'énergie à la fin d'une période à partir du niveau d'énergie de la période précédente, de la quantité d'énergie acquise en se rechargeant et les la quantité dépensée soit en état de veille, soit en exécutant les missions affectées au robot durant la période en considération. Les contraintes (4.23) garantissent que la quantité d'énergie acquise par un robot en état de rechargement ne dépasse pas la vitesse de rechargement du chargeur. Les contraintes (4.24) vérifient que le niveau d'énergie emmagasiné sur les robots ne dépasse pas les bornes inférieures et supérieures recommandées. Les contraintes (4.25) vérifient la disponibilité d'un nombre minimal  $L_t$  de robots opérationnels. Les contraintes (4.26) représentent les contraintes d'intégrité des variables de décisions.

Le problème maître (PM) est constitué d'un très grand nombre de colonnes, compliquant la résolution directe de ce problème en nombres entiers. De ce fait, on procède à une relaxation des contraintes d'intégrité (4.26) en les remplaçant par les contraintes suivantes :  $0 \leq \lambda_p \leq 1; 0 \leq u_{mt} \leq 1$ . Le problème relaxé obtenu est connue sous le nom « Problème Maître Linéaire » (PML) et ce dernier est résolu en utilisant les méthodes de génération de colonnes. Sachant l'amplitude de l'espace de résolution de problème, on considère une partie de l'espace de calcul en démarrant avec un sous ensemble  $\Omega' \subset \Omega$  de colonnes. Ce problème est connu sous le nom du « Problème Maître Restreints » (PMR) et des nouvelles colonnes sont générées au fur et au mesure, en utilisant des méthodes de Simplex.

Afin de trouver une solution pour le problème PML par génération de colonnes, on démarre avec un PMR avec un nombre réduit de colonnes et le problème PMR est résolu en utilisant la méthode de Simplex qui permet de déterminer la solution optimale du PMR et les valeurs des variables duales du problème PML. Le problème de pricing consiste à déterminer les colonnes ayant les coûts réduits minimaux. Si ces coûts réduits minimaux sont non négatifs, alors la solution optimale du PML est trouvée. Sinon, les colonnes ayant les coûts réduits négatifs sont ajoutées au sous ensemble  $\Omega'$  et le processus est répété tant qu'on trouve des colonnes avec des coûts réduits négatifs.

Soient  $\mu_{nt}, \gamma_{mt}, \pi_{ft}, \eta_{ht}, \nu_{nt}, \tau_{nt}, \delta_t$  les valeurs des variables duales de la solution optimale du PMR correspondant respectivement aux contraintes (4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.25).

Le coût réduit d'une colonne relatif au robot  $n$  en période  $t$  est donné par la formule suivante :

$$\begin{aligned}
PP(n,t) = & \min \sum_{h \in H} \sum_{m \in M} \phi_{m,t} (\alpha A_{mh} + T_m + \beta R_{mh}) X_h^p Z_m^p - \mu_{nt} - \sum_{m \in M} \gamma_{mt} Z_m^p \\
& - \sum_{f \in F} \pi_{ft} \sum_{c \in C} a_{cf} Y_c^p - \sum_{h \in H} \eta_{ht} X_h^p - \sum_{m \in M} \phi_{mt} e_m Z_m^p v_{nt} \\
& - l_n v_{nt} + W_{nt} v_{nt} + G_n W_{nt} \tau_{nt} - W_{nt} \delta_t
\end{aligned}$$

Afin de réduire la complexité du problème de pricing, on le décompose en plusieurs sous problèmes dont chacun concerne une station d'attente  $h$  et une configuration  $c$ . Pour chaque couple  $(h,c)$ , on détermine l'ensemble des missions qui génèrent le coût réduit minimum. Plus précisément le sous- problème de pricing consiste à déterminer l'ensemble des missions qui minimisent la formule suivante :

$$\begin{aligned}
SPP(n,t;h,c) = & \min \sum_{m \in M} \phi_{m,t} (\alpha A_{mh} + T_m + \beta R_{mh}) Z_m^p - \mu_{nt} - \sum_{m \in M} \gamma_{mt} Z_m^p \\
& - \sum_{f \in F} \pi_{ft} a_{cf} - \phi_{ht} - \sum_{m \in M} \phi_{m,t} e_m v_{nt} Z_m^p
\end{aligned}$$

Ce qui est équivalent à :

$$SPP(n,t;h,c) = \sum_{m \in M} [\phi_{m,t} (\alpha A_{mh} + T_m + \beta R_{mh}) - \gamma_{mt} - \phi_{m,t} e_m v_{nt}] Z_{nmt}^p - const \quad (4-28)$$

Sous les contraintes:

$$\sum_{m \in M} \phi_{m,t} (A_{mh} + T_m + R_{mh}) Z_m^p \leq T_{t,max} \quad (4-29)$$

$$Z_m^p = 0 \quad \text{if } c \notin C_m \quad (4-30)$$

$$Z_m^p \in \{0,1\} \quad \forall m \in M \quad (4-31)$$

Notons que les sous- problèmes de pricing  $SPP(n,t;h,c)$  sont équivalents aux problèmes de sac à dos (knapsack problem), et peuvent être résolus en utilisant les algorithmes de programmation dynamique pseudo-polynomiaux [Martello 90] afin de déterminer pour chaque sous problème  $PP(n,t)$  le couple  $(h,c)$  résultant de la colonne ayant le cout réduit le plus bas.

### 4.3. Planification stratégique en deux phases

Au cours de la deuxième année du projet IWARD, le groupe de travail Hardware a divulgué son intention de développer et d'annexer au robot un système de gestion d'autonomie qui utilise des batteries amovibles assurant la continuité du travail en hôpital. Ce qui a nécessité la dissociation du problème de gestion d'autonomie de celui de la gestion des configurations et de la sélection des stations d'attentes.

Pour cela, nous proposons de planifier le rechargement des robots ; en premier lieu si les robots ne sont pas équipés de batteries amovibles et ensuite de planifier conjointement la configuration des robots et leur localisation.

L'agent de décision responsable de la gestion des configurations et de la sélection des stations d'attente doit être assez flexible. Il doit prendre en compte les décisions prises par l'agent de gestion d'autonomie dans le cas où les robots se rechargent durant des longues périodes. Dans le cas où les robots sont équipés de batteries amovibles, la problématique de l'autonomie ne

sera pas évoquée et n'est pas considérée dans le calcul de gestion des configurations et de localisation.

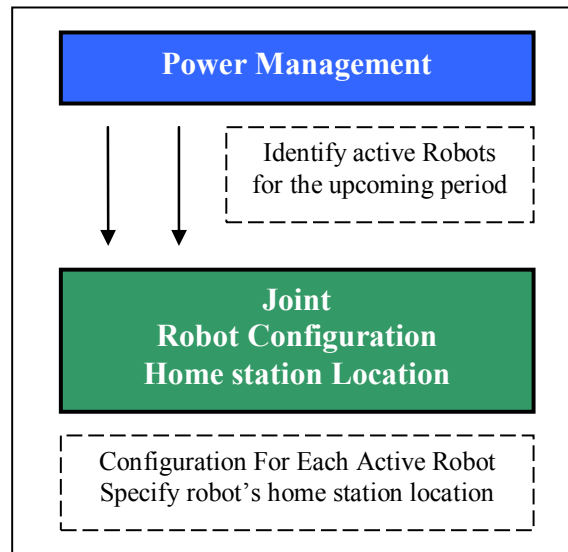


Figure 15 : Planification stratégique en deux phases

#### 4.3.1. Modèle mathématique du problème de gestion d'énergies

Le problème de gestion d'autonomie des robots concerne un système robotique composé d'un ensemble  $N$  de robots équipés de batteries fixes ayant une capacité énergétique limitée et dont la durée de rechargement est assez importante.

Les hypothèses suivantes sont considérées :

- Les rechargements partiels sont interdits pour préserver la durée de vie des batteries et les robots ne peuvent pas changer d'état de chargement durant une période. Cette hypothèse exclut le cas où les robots se chargent dans les stations d'attente équipées des chargeurs automatiques.
- Les configurations des robots ne sont pas prises en compte et les missions sont affectées aux robots sans tenir compte des configurations nécessaires.
- La localisation n'étant pas prise en compte, nous ne considérons pas les contraintes des charges de travail des robots.
- Les décisions tactiques comme l'affectation et l'ordonnancement des missions en temps réel ne sont pas prise en compte à ce stade de planification.
- Les courbes de chargement et déchargement sont considérées linéaires entre deux bornes supérieures et inférieures.

Le but de cet agent de décision est d'optimiser le nombre des robots opérationnels durant la journée afin d'avoir des robots de secours pouvant faire face à des pics de demandes occasionnelles, et de maximiser le nombre de missions prises en charge par le système robotique.

On considère que chaque journée est divisée en plusieurs créneaux  $t \in T$  et on définit par  $\phi_{mt}$  la fréquence d'occurrence de mission  $m$ , durant le créneau  $t$ .

Chaque robot  $n$  est doté d'une batterie. Soit  $Q_{n,0}$  le niveau d'énergie initial du robot  $n$ . Une quantité  $l_n$  d'énergie est consommée par le robot  $n$  en état opérationnel et une quantité additionnelle  $e_m$  d'énergie est nécessaire pour chaque mission  $m$  exécutée. Le rechargement d'un robot  $n$  dans un créneau, permet à ce dernier d'acquérir au maximum une quantité  $G_n$  d'énergie.

Afin de modéliser cet agent en un programme linéaire en nombre entiers, les variables de décisions suivantes sont envisagées :

$$v_{n,t} = \begin{cases} 1, & \text{si le robot } n \text{ est opérationnel durant le créneau } t \\ 0, & \text{si le robot } n \text{ se recharge durant le créneau } t \end{cases}$$

$$z_{nm,t} = \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot } n, \text{ durant le créneau } t \\ 0, & \text{sinon} \end{cases}$$

$$u_{m,t} = \begin{cases} 0, & \text{si la mission } m \text{ pris en charge par un robot, durant le créneau } t \\ 1, & \text{sinon} \end{cases}$$

Le problème peut être formulé ainsi

$$\text{Max} \sum_{n \in N} \sum_{t \in T} v_{n,t} - \sum_{m \in M} P_m \phi_{m,t} u_{m,t} \quad (4-32)$$

Sous contraintes :

$$u_{m,t} + \sum_n z_{nm,t} = 1 \quad \forall m \in M; t \in T \quad (4-33)$$

$$\sum_n v_{n,t} \geq L_t \quad \forall t \in T \quad (4-34)$$

$$z_{nm,t} \leq v_{n,t} \quad \forall n \in N; t \in T; m \in M \quad (4-35)$$

$$q_{n,t} = q_{n,t-1} - l_n v_{n,t} - \sum_{m \in M_t} \phi_{m,t} e_m z_{nm,t} + g_{n,t}, \forall n \in N; t \in T \quad (4-36)$$

$$\underline{Q}_n \leq q_{n,t} \leq \bar{Q}_n \quad \forall n \in N; t \in T \quad (4-37)$$

$$0 \leq g_{n,t} \leq G_n (1 - v_{n,t}) \quad \forall n \in N; t \in T \quad (4-38)$$

$$v_{n,t}, z_{nm,t} \in \{0;1\}, q_{n,t}, g_{n,t} \in \mathbb{R} \quad \forall n \in N; t \in T; m \in M \quad (4-39)$$

La fonction objectif (4.32) a pour but de maximiser le nombre des robots opérationnels, et de maximiser le nombre de missions assignées aux robots ce qui est équivalent à minimiser le nombre de missions rejetées. Les contraintes (4.33) vérifient que toute mission est soit affectée à un robot soit rejetée. Les contraintes (4.34) vérifient que le nombre de robots en état d'opérationnels est supérieur au nombre  $L_t$  de robots garantissant un service minimum durant la période  $t$ . Les contraintes (4.35) vérifient que les missions sont affectées à des robots opérationnels. Les contraintes (4.36) détermine le niveau d'énergie à la fin d'une période à partir du niveau d'énergie de la période précédente, de la quantité d'énergie acquise en se rechargeant et la quantité dépensée soit en état de veille, soit en exécutant les missions affectées au robot durant la période en considération. Les contraintes (4.37) vérifient que le

niveau d'énergie emmagasiné sur les robots ne dépasse pas les bornes inférieures et supérieures recommandées. Les contraintes (4.38) garantissent que les robots opérationnels ne peuvent pas être rechargés partiellement et que la quantité d'énergie acquise par un robot en état de rechargement ne dépasse pas la vitesse de rechargement du chargeur. Les contraintes (39) représentent les contraintes d'intégrité des variables de décisions.

Après la résolution de ce modèle à l'aide d'un solveur MIP, on retient pour chaque robot son état au premier créneau  $(v_{n,0})$ , en plus pour les opérationnels  $(v_{n,0}=I)$ , on calcule  $E_n$  la quantité d'énergie recommandée pour le robot  $n$  qui est donnée par la formule suivante :

$$E_n = \sum_{m \in M} \phi_{m,t} e_m Z_{nm,t}.$$

Cependant lors de la seconde phase de calcul, compte tenu du fait que certaines missions ne peuvent pas être exécutées sur certains robots faute de configurations appropriées, on tolère un dépassement par rapport à la recommandation initiale à condition de respecter les limites minimales physiques de la batterie. Et soit  $E_{n,max}$  la quantité maximale que le robot  $n$  peut dépenser,  $E_{n,max}$  est calculée selon la formule suivante :  $E_{n,max} = Q_{n,0} - \underline{Q}_n - I_n$ .

#### 4.3.2. Planification intégrée de la configuration et de la localisation des robots

Le modèle mathématique doit prendre en compte les recommandations énergétiques données par le modèle précédent, et relaxer ces contraintes dans le cas où les robots sont équipés par des batteries amovibles.

Le but est de déterminer pour chaque robot  $n$  opérationnel, sa station d'attente par défaut et les fonctionnalités qui seront installées sur le robot pour la période à venir.

Rappelons que le système robotique est caractérisé par les notations suivantes :

- $F$  ensemble des fonctionnalités
- $N_f$  nombre d'exemplaires disponibles de la fonctionnalité  $f$
- $C$  ensemble de configurations possibles d'un robot
- $a_{fc}$  nombre d'exemplaires de la fonctionnalité  $f$  dans la configuration  $c$
- $H$  ensemble des stations d'attente.

Chaque mission  $m$  peut être caractérisée par les paramètres suivants:

- $\phi_m$  fréquence d'occurrence de la mission  $m$
- $e_m$  quantité d'énergie nécessaire pour exécuter la mission  $m$
- $T_m$  temps nécessaire pour exécuter la mission  $m$
- $C_m$  ensemble des configurations permettant l'exécution de la mission  $m$
- $A_{hm}$  temps nécessaire pour un robot d'effectuer le trajet de la station d'attente  $h$  jusqu'au point de départ de la mission  $m$
- $R_{hm}$  temps nécessaire pour un robot d'effectuer le trajet du retour du point de terminaison de la mission  $m$  jusqu'à la station d'attente  $h$ .

Les variables de décisions suivantes sont employées :

$$\begin{aligned}
x_{nh} &= \begin{cases} 1, & \text{si le robot } n \text{ est affecté à la station d'attente } h \\ 0, & \text{sinon.} \end{cases} \\
y_{nc} &= \begin{cases} 1, & \text{si le robot } n \text{ utilise la configuration } c \\ 0, & \text{sinon.} \end{cases} \\
z_{nm} &= \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot } n \\ 0, & \text{sinon.} \end{cases} \\
u_m &= \begin{cases} 0, & \text{si la mission } m \text{ n'est pas prise en charge par un robot} \\ 1, & \text{sinon.} \end{cases} \\
r_{nhm} &= \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot } n \text{ stationné en } h \\ 0, & \text{sinon.} \end{cases}
\end{aligned}$$

La consommation d'énergie recommandée  $E_n$  peut être dépassée. Soit  $s_n$  le dépassement de cette recommandation et soit  $\chi$  la pénalité de ce dépassement.

Le problème combiné de configuration et localisation est formulé ainsi :

$$\text{Minimize } \sum_{n \in N} \sum_{h \in H} \sum_{m \in M} \phi_m (\alpha A_{hm} + T_m + \beta R_{hm}) r_{nhm} + \sum_{n \in N} \chi s_n - \sum_{m \in M} p_m u_m \quad (4-40)$$

Sous contraintes:

$$\sum_{h \in H} x_{nh} = 1, \quad \forall n \in N \quad (4-41)$$

$$\sum_{c \in C} y_{nc} = 1, \quad \forall n \in N \quad (4-42)$$

$$\sum_{n \in N} z_{nm} + u_m = 1, \quad \forall m \in M \quad (4-43)$$

$$\sum_{c \in C} \sum_{n \in N} a_{fc} y_{nc} \leq N_f, \quad \forall f \in F \quad (4-44)$$

$$\sum_{n \in N} x_{nh} \leq H_h, \quad \forall h \in H \quad (4-45)$$

$$\sum_{h \in H} \sum_{m \in M} \phi_m (A_{hm} + T_m + R_{hm}) r_{nhm} \leq T_{\max}, \quad \forall n \in N \quad (4-46)$$

$$\sum_{c \in C_m} y_{nc} \geq z_{nm}, \quad \forall m \in M, \forall n \in N \quad (4-47)$$

$$x_{nh} + z_{nm} \leq 1 + r_{nhm} \quad \forall h \in H, \forall m \in M, \forall n \in N \quad (4-48)$$

$$\sum_{m \in M} e_m z_{nm} \leq E_n + s_n \quad \forall n \in N \quad (4-49)$$

$$E_n + s_n \leq E_{n,\max} \quad \forall n \in N \quad (4-50)$$

$$x_{nh}, y_{nc}, z_{nm}, r_{nhm} \in \{0,1\}, s_n \geq 0; \quad \forall h \in H, \forall m \in M, \forall n \in N, c \in C \quad (4-51)$$

La fonction objectif (4.40) a pour but de minimiser le dépassement des recommandations énergétiques des robots ainsi que la charge de travail de l'ensemble des robots. Les contraintes (4.41) vérifient que chaque robot opérationnel est affecté à une station d'attente,



les contraintes (4.42) vérifient que chaque robot opérationnel est associé à une configuration. Les contraintes (4.43) assurent que toute mission est affectée à un et un seul robot. Les contraintes (4.44) garantissent que le nombre des fonctionnalités utilisées et installées sur les robots ne dépasse pas le nombre de fonctionnalités existantes. Les contraintes (4.45) garantissent que le nombre de robots affectés à une station d'attente ne dépasse pas les limites prédéfinies. Les contraintes (4.46) garantissent que la charge maximale de travail d'un robot n'est pas dépassée. Les contraintes (4.47) certifient qu'une mission n'est affectée qu'au robot convenablement configuré. Les contraintes (4.48) assurent la linéarisation du produit  $z_{nm}x_{nh}$  en utilisant la variable  $r_{nhm}$ . Les contraintes (4.49) déterminent le dépassement de la quantité d'énergie recommandée. Les contraintes (4.50) garantissent les contraintes des limites physiques des batteries. Les contraintes (4.51) représentent les contraintes d'intégrité des variables de décisions.

Dans le cas où ce modèle se révèle sans solution applicable, les contraintes (4.43) sur l'affectation obligatoire des missions sont remplacées par les contraintes (4.33) pour autoriser la non affectation d'une mission. Les coûts de non affectation doivent alors être introduits dans (4-40).

Le cas où les robots sont équipés de batteries amovibles, implique l'omission des contraintes reliées à la gestion de l'autonomie des batteries donc les contraintes (4.49, 4.50) ainsi que les pénalités sur les dépassements énergétiques de la fonction objective (4.40)

Les expérimentations numériques ont révélé certaines limites par rapport à la taille des instances. Nous proposons ainsi une approche de génération de colonnes.

Le modèle précédent peut être rendu plus compact, si pour toute station  $h$ , au plus un robot est stationné dans une station d'attente, i.e.  $H_h = 1$ . Dans ce cas, la variable  $z_{nm}$  est remplacée par  $z_{hm} = 1$  si la mission  $m$  est affectée au robot affecté à la station  $h$  et les variables  $r_{nhm}$  de linéarisation ne sont pas nécessaires. Le modèle devient alors:

$$\text{Minimize } \sum_{h \in H} \sum_{m \in M} \phi_m (\alpha A_{hm} + T_m + \beta R_{hm}) z_{hm} + \sum_{h \in H} \chi s_h - \sum_{m \in M} p_m u_m \quad (4-52)$$

Sous contraintes:

$$\sum_{h \in H} x_{nh} = 1, \quad \forall n \in N \quad (4-53)$$

$$\sum_{c \in C} y_{hc} \leq 1, \quad \forall h \in H \quad (4-54)$$

$$\sum_{h \in H} z_{hm} + u_m = 1, \quad \forall m \in M \quad (4-55)$$

$$\sum_{c \in C} \sum_{h \in H} a_{jc} y_{hc} \leq N_f, \quad \forall f \in F \quad (4-56)$$

$$\sum_{n \in N} x_{nh} \leq 1, \quad \forall h \in H \quad (4-57)$$

$$\sum_{m \in M} \phi_m (A_{hm} + T_m + R_{hm}) z_{hm} \leq T_{\max}, \quad \forall n \in N \quad (4-58)$$

$$\sum_{c \in C_m} y_{hc} \geq z_{hm}, \quad \forall m \in M, \forall h \in H \quad (4-59)$$

$$\sum_{m \in M} e_m z_{hm} \leq \sum_{n \in N} x_{nh} E_n + s_h \quad \forall h \in H \quad (4-60)$$

$$\sum_{n \in N} x_{nh} E_n + s_h \leq \sum_{n \in N} x_{nh} E_{n, \max} \quad \forall h \in H \quad (4-61)$$

$$x_{nh}, y_{nc}, z_{hm} \in \{0, 1\}, s_h \geq 0; \quad \forall h \in H, \forall m \in M, \forall n \in N, c \in C \quad (4-62)$$

Les autres modèles de ce chapitre peuvent également être simplifiés de manière similaire.

### 4.3.3. Génération de colonnes pour la planification intégrée de la configuration et de la localisation

Vue la complexité de ce problème, nous développons des heuristiques et des méthodes permettant de résoudre des instances de taille réelle de ce problème. Dans cette section nous présentons d'abord une formulation de ce problème en génération de colonnes, les méthodes de pricing pour la génération des nouvelles colonnes, permettant de déterminer une borne inférieure du problème. Ensuite on génère la solution entière en utilisant les méthodes d'arrondissement arithmétiques.

Une reformulation du problème utilisant le principe de génération de colonnes a été élaboré pour la planification intégrée et les notations utilisées restent valables pour cette section. Dans cette reformulation, une colonne représente un robot affecté à une station d'attente, configuré selon une configuration donnée et l'ensemble des missions affectées.

Soit  $\Omega$  l'ensemble des colonnes faisables, sachant qu'une colonne  $p$  est représentée par les paramètres suivants :

$$X_{ph} = \begin{cases} 1, & \text{si la colonne } p \text{ concerne la station d'attente } h \\ 0, & \text{sinon.} \end{cases}$$

$$Y_{pc} = \begin{cases} 1, & \text{si la colonne } p \text{ est reliée à la configuration } c \\ 0, & \text{sinon.} \end{cases}$$

$$Z_{pm} = \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot de la colonne } p \\ 0, & \text{sinon.} \end{cases}$$

Une colonne est dite faisable si elle est affectée à une seule station d'attente et une seule configuration telle que les missions affectées au robot puissent être accomplies par les fonctionnalités installées sur le robot, et que la charge de travail respecte les limites prédéfinies  $T_{max}$ .

Pour une colonne  $p$  donnée, la charge/coût peut être déterminée suivant la formule suivante :

$$b_p = \sum_{h \in H} \sum_{m \in M} \phi_m (A_{hm} + T_m + R_{hm}) X_{ph} Z_{pm} \quad (4-63)$$

avec l'hypothèse  $\alpha = \beta = 1$ .

Une seule variable de décision est utilisée:

$$\lambda_p = \begin{cases} 1, & \text{si la colonne } p \text{ est selectionnée} \\ 0, & \text{sinon.} \end{cases}$$

Ainsi le problème intégré de localisation et de configuration est reformulé comme un «Problème Maître» (PM) :

$$\text{Minimiser } \sum_{p \in \Omega} b_p \lambda_p \quad (4-64)$$

Sous contraintes:

$$\sum_{p \in \Omega} \lambda_p = N \quad (4-65)$$

$$\sum_{p \in \Omega} Z_{pm} \lambda_p = 1, \quad \forall m \in M \quad (4-66)$$

$$\sum_{c \in C} \sum_{p \in \Omega} a_{fc} Y_{pc} \lambda_p \leq N_f \quad \forall f \in F \quad (4-67)$$

$$\sum_{p \in \Omega} X_{ph} \lambda_p \leq H_h, \quad \forall h \in H \quad (4-68)$$

$$\lambda_p \in \{0, 1\} \quad \forall p \in \Omega \quad (4-69)$$

La fonction objectif (4.64) a pour but de minimiser la charge de travail du système robotique. Les contraintes (65) assurent que tous les  $N$  robot sont considérés et affectés. Les contraintes (4.66) garantissent que toutes les missions sont prises en charge par un robot. Les contraintes (4.67) certifient que le nombre des fonctionnalités à installer sur les robots est inférieur ou égal au nombre d'exemplaires disponibles. Les contraintes (4.68) assurent que le nombre de robots affectés à une station d'attente respecte les limites prédéfinies de ce que la station peut accueillir. Les contraintes (4.69) sont des contraintes d'intégrité.

Le problème maître (PM) contient un très grand nombre de colonnes et ne peut pas être résolu directement. On procède à une relaxation des contraintes d'intégrités (4.69) qui seront remplacées par les contraintes suivantes :  $\lambda_p \geq 0$  ( $\forall p \in \Omega$ ) et le problème obtenu, sera désigné « problème maître linéaire » (PML). Vu le grand nombre des colonnes dans ce modèles, une autre variante de ce modèle dénommée «Problème Maître Restreint» (PMR) limité à un sous ensemble des solutions possibles  $\Omega' \subset \Omega$  est considérée, et les colonnes pouvant améliorer la solution sont générées au fur et à mesure.

## Génération de colonnes

Afin de trouver la solution optimale du PML, on commence par la résolution du PMR jusqu' à l'optimalité en utilisant l'algorithme du Simplex. Cela conduit à la solution optimale du PMR ainsi qu'aux valeurs des variables duales du problème PML. Les variables duales permettent la détermination des coûts réduits des variables et ainsi la génération des colonnes améliorantes.

Soit  $u$ ,  $v_m$ ,  $w_f$  et  $r_h$  les variables duales du problème PMR correspondant respectivement pour les contraintes (4.65, 4.66, 4.67, 4.68).

Le principe de génération de colonnes (pricing problem) consiste à déterminer les colonnes ayant les plus bas coûts réduits. Si les coûts réduits des colonnes obtenues sont non négatifs, la solution optimale du problème PML est obtenue. Sinon des colonnes avec des coûts réduits négatifs sont ajoutées à l'ensemble  $\Omega'$  et le processus est répété jusqu'à l'optimalité du PML.

Afin de réduire la complexité du problème de pricing PB, on le décompose en sous problèmes. Chacun de ces sous problèmes est relié à une station d'attente  $h$  et à une configuration  $c$ . Chaque sous problème  $PB(h,c)$  permet de déterminer la colonne ayant le plus bas coût réduit  $\pi_{hc}$ .

Le sous problème de pricing **SPP(h,c)** consiste à minimiser le coût réduit parmi les colonnes correspondant à (h, c) :

$$\pi_{hc} = \text{Minimize } b_p - u - \sum_{m \in M} v_m Z_{pm} - \sum_{f \in F} a_{fc} w_f - r_h$$

Ce qui est équivalent à :

$$\pi_{hc} = \text{Minimize } \sum_{m \in M} (\phi_m (A_{hm} + T_m + R_{hm}) - v_m) Z_{pm} - u - \sum_{f \in F} a_{fc} w_f - r_h \quad (4-70)$$

Sous contraintes:

$$\sum_{m \in M} \phi_m (A_{hm} + T_m + R_{hm}) Z_{pm} \leq T_{\max} \quad (4-71)$$

$$Z_{pm} = 0, \text{ si } c \notin C_m \quad (4-72)$$

$$Z_{pm} \in \{0,1\}, \forall m \in M \quad (4-73)$$

Le sous problème SPP est un problème de la famille de sac à dos, qui peut être résolu en appliquant un algorithme de programmation dynamique pseudo polynomial [Martello 90].

### Solution initiale et variables artificielles

Afin de résoudre le problème PML, il est nécessaire que le premier sous ensemble de  $\Omega'$  contienne assez de colonnes constituant une solution faisable, ce qui permet de calculer les valeurs duales des contraintes et générer les colonnes.

Deux solutions sont envisageables : soit le développement des heuristiques permettant de construire des solutions faisables, soit la modification du modèle et l'introduction des variables artificielles permettant de trouver une solution pour le modèle même sans solution initiale.

Dans notre étude, la seconde approche a été choisie. Pour chacune des contraintes du modèle PML, une variable artificielle est injectée, et l'usage de ces variables artificielles est pénalisé, par l'ajout d'une pénalité de  $(2 \times T_{\max})$  à la fonction objectif. Ainsi le PML peut être remplacé par le modèle suivant :

$$\text{Minimizer } \sum_{p \in \Omega} b_p \lambda_p + 2 \times T_{\max} \left( P + \sum_{m \in M} Q_m \right) \quad (4-74)$$

Sous contraintes:

$$\sum_p \lambda_p + P = N \quad (4-75)$$

$$\sum_{p \in \Omega} Z_{pm} \lambda_p + Q_m = 1 \quad \forall m \in M \quad (4-76)$$

$$\sum_{c \in C} \sum_{p \in \Omega} a_{jc} Y_{pc} \lambda_p + Q_f \leq N_f, \forall f \in F \quad (4-77)$$

$$\sum_{p \in \Omega} X_{pb} \lambda_p + Q_h \leq H_h \quad \forall h \in H \quad (4-78)$$

$$\lambda_p \geq 0, \forall p \in \Omega, P \geq 0, Q \geq 0 \quad (4-79)$$

La solution  $P = N$ ,  $Q_m = 1$ ,  $Q_f = 0$ ,  $Q_h = 0$ , et  $\lambda_p = 0$  est toujours faisable, et a une pénalité ( $2 \times T_{max}$ ) des variables artificielles qui est supérieure aux coûts de toutes les colonnes faisables de l'ensemble  $\Omega$ . Ces variables artificielles seront sorties de base lors des itérations suivantes de générations de colonnes.

### Construction d'une solution admissible

Après la convergence de la procédure de génération de colonnes, la solution optimale du problème PML est obtenue. Sachant que le problème PML est obtenu par la relaxation des contraintes d'intégrité (69) du problème maître PM, PML fournit une borne inférieure du problème maître mais la solution obtenue peut contenir des valeurs fractionnelles des variables  $\lambda_p$  et ainsi une solution non intégrale du PML n'est pas une solution faisable de PM.

Plusieurs approches peuvent être proposées pour déduire une solution au problème PM de la solution du PML. Une approche consiste à appliquer les procédures de Branch and Price pour déterminer une solution du problème maître.

Dans notre étude, nous utilisons une heuristique qui consiste à fixer à 1 les colonnes ayant des valeurs intégrales ( $\lambda_p = 1$ ) et d'arrondir à 1 la colonne ayant une valeur fractionnelle la plus élevée. Une fois que ces valeurs sont fixées, on les retire du problème pour obtenir un problème plus réduit en robots et en missions. Ce processus est répété tant qu'on obtient une solution optimale fractionnelle du PML.

### Vue d'ensemble de la procédure de génération de colonnes

En résumé, plusieurs étapes et procédures sont nécessaires pour pouvoir résoudre notre problème de localisation et configuration à l'aide de l'approche de génération de colonne. La *Figure 16* illustre les étapes-clés des techniques de génération de colonnes.

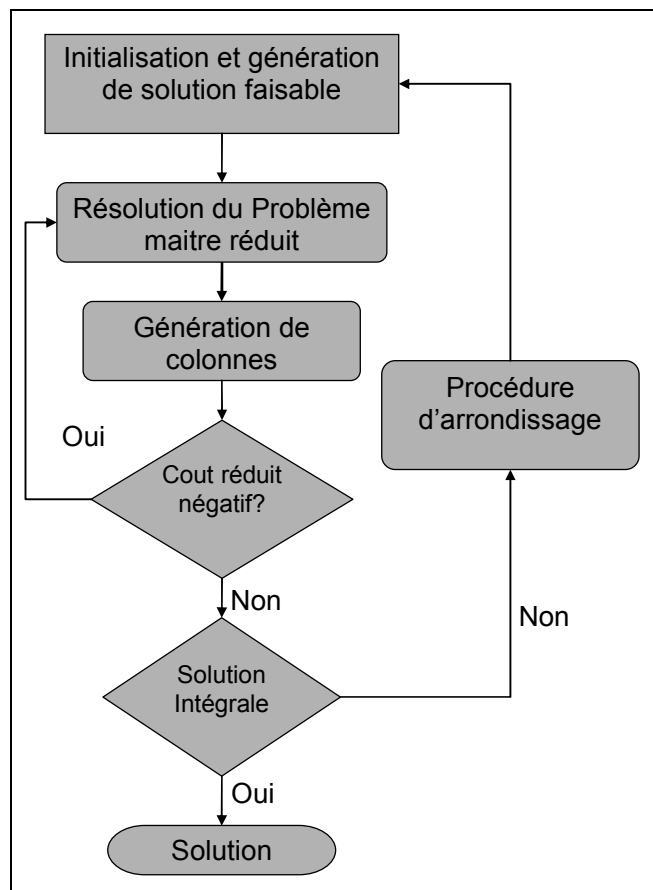


Figure 16 : Approche de génération de colonnes

La première étape consiste à transformer notre problème en un problème Maître Linéaire, ensuite on procède à la génération d'une solution afin de pouvoir construire le premier sous ensemble de colonnes du Problème Maître Restreint. Dans notre étude on propose d'ajouter des variables artificielles permettant de générer une solution évidente pour le PML.

En deuxième étape, à l'aide d'un solveur standard de programmation linéaire, on résout le Problème Maître restreint (PMR) qui contient un nombre restreint de colonnes générées précédemment.

En troisième étape, on procède à la génération de colonnes en utilisant la programmation dynamique et les valeurs des variables duales du problème PMR. Les colonnes générées ayant un coût réduit négatif sont ajoutées à l'espace de colonnes du PMR.

Les étapes 2 et 3 sont répétées tant qu'on peut trouver des colonnes ayant un coût réduit négatif. Si toutes les colonnes générées ont des coûts réduits positifs, la solution trouvée du PMR est la solution optimale du PML. Si la solution obtenue est intégrale, la solution optimale du problème Maître (PM) est obtenue.

La quatrième étape a pour but de trouver une solution intégrale pour le problème maître. Pour cela on fixe à 1 les colonnes égales à 1 et la colonne ayant la valeur fractionnelle la plus grande et on les retire du problème pour construire un sous problème maître réduit et les étapes 2, 3 et 4 sont répétées tant qu'une solution intégrale n'est pas trouvée.

### Instance illustrative

Dans cette section on illustre l'approche de génération de colonnes par une instance du problème combiné de configuration et de localisation. Le problème considéré est illustré par la Figure 17. Il y a trois stations d'attente  $\{H1, H2, H3\}$ , deux robots de base  $\{R1, R2\}$  dont la charge de travail maximale  $T_{max}$  est de 150 minutes (2.5 heures). Deux types de fonctionnalités sont considérés: les fonctionnalités de transport et les fonctionnalités de nettoyage. On dispose de deux modules de transport  $N_{transp} = 2$  et une module de nettoyage  $N_{clean} = 1$ .

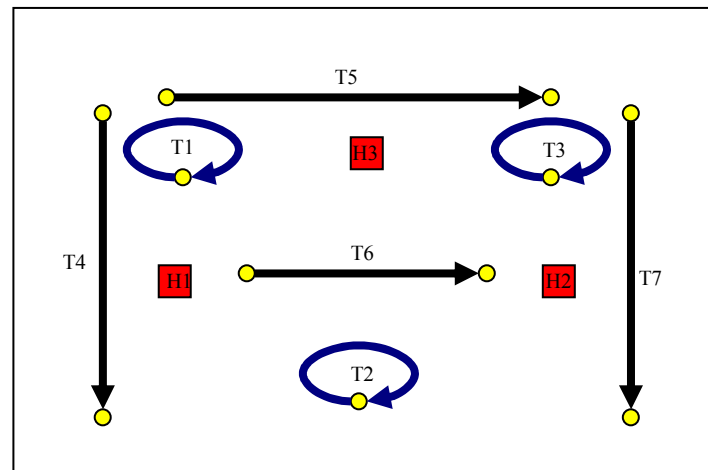


Figure 17 : Instance de 7 missions et 3 stations d'attente

Sept missions sont prévues d'être exécutées. Les caractéristiques de ces missions sont présentées dans le tableau 3, notamment le type de la mission et le temps nécessaires pour accomplir une mission en partant d'une station d'attente donnée. Ce temps comprend les temps de réaction, le temps d'exécution et le temps de retour. La fréquence d'occurrence des missions  $f_m$  est égale à 1 pour toutes les missions, et le nombre maximal  $H_h$  de robots qu'une station d'attente peut accueillir est 1. Il y a trois configurations possibles pour chaque robot :  $C = \{\{clean\}, \{transp\}, \{clean, transp\}\}$  ce qui se traduit par qu'un robot est équipé, soit d'une fonctionnalité  $\{clean\}$  ou  $\{transp\}$ , soit des deux fonctionnalités  $\{clean, transp\}$  en même temps.

Mission	Type	Temps nécessaire pour exécuter la mission en démarrant de:		
		H1	H2	H3
M1	<i>clean</i>	20	32	20
M2	<i>clean</i>	24	24	30
M3	<i>clean</i>	32	20	20
M4	<i>transp</i>	26	42	35
M5	<i>transp</i>	40	40	31
M6	<i>transp</i>	15	15	17
M7	<i>transp</i>	42	26	35

Tableau 3: Caractéristiques des missions

Cette instance est résolue par l'algorithme de génération de colonnes présentées précédemment. La solution obtenue est exposée dans le Tableau 4: Solution de l'instance illustrative avec  $T_{max}=150$

. On note que cette solution est une solution optimale. Pour chaque robot il existe une

station d'attente à laquelle il est assigné, sa configuration et les missions qui lui seront affectées ainsi que la charge résultante.

Robot	Station	Configuration	Missions	Charge (min)
R1	H2	{ <i>transp</i> }	{T6, T7}	41
R2	H3	{ <i>clean, transp</i> }	{T1, T2, T3, T4, T5}	136

Tableau 4: Solution de l'instance illustrative avec  $T_{max}=150$

A partir des résultats du Tableau 4: Solution de l'instance illustrative avec  $T_{max}=150$ , on constate que la charge totale du système robotique est de 177 minutes mais elle n'est pas répartie de manière équilibrée sur les deux robots. La Figure 18 représente la solution présentée par le Tableau 4

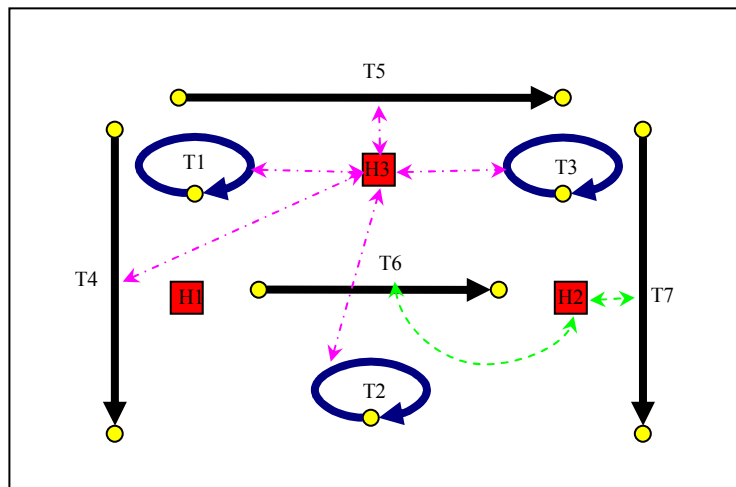


Figure 18 : Solution de l'instance illustrative

En réduisant la charge maximale des robots de 150 minutes à 110 minutes, la solution obtenue par la même procédure de génération de colonne est mieux équilibrée. Le tableau 5 représente la solution de cette instance modifiée.

Robot	Station	Configuration	Missions	Charge(min)
R1	H1	{ <i>transp</i> }	{T4, T5, T6}	81
R2	H2	{ <i>clean, transp</i> }	{T1, T2, T3,	102

Tableau 5: Solution de l'instance illustrative avec  $T_{max}=110$

La charge totale du système a augmenté de 177 min à 188 min ce qui représente des parcours plus longs. Par contre les charges de travail sont bien mieux équilibrées.

#### 4.4. Planification stratégique séquentielle en trois phases

La troisième approche de la planification stratégique consiste à résoudre les trois problèmes stratégiques de manière séquentielle, en prenant en compte les résultats de l'étape précédente. En premier lieu le problème de l'autonomie des batteries des robots est abordé et on définit pour chaque robot un plan de rechargement, ce qui permet de déterminer les robots opérationnels pour les étapes suivantes. Ensuite, après l'identification des robots opérationnels pour la période suivante, nous abordons le problème de configuration des robots pour déterminer les fonctionnalités de robots afin d'équilibrer la charge des robots. A la fin,



connaissant la configuration des robots pour la période suivante, les robots seront assignés aux stations d'attente les plus proches des missions qu'ils peuvent accomplir avec les fonctionnalités aménagées sur les robots.

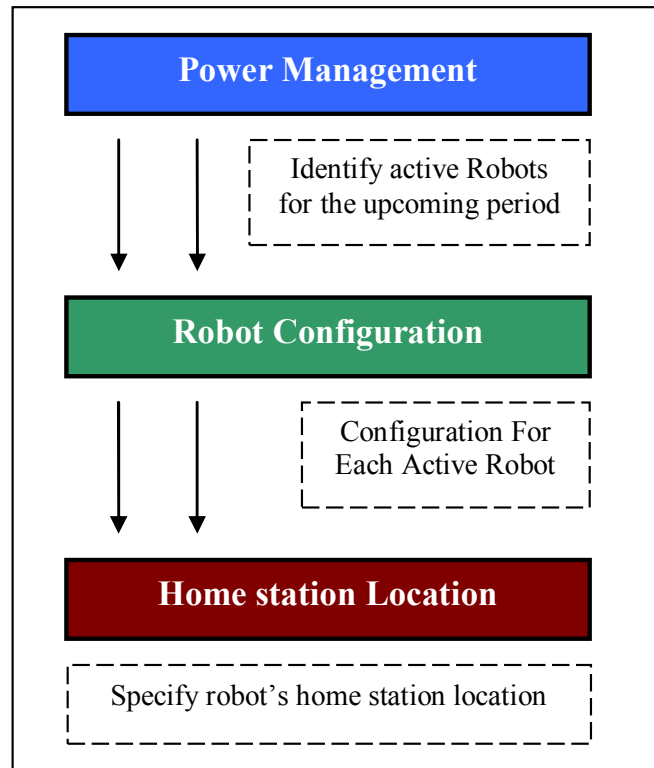


Figure 19 : Planification stratégique en trois phases

Le modèle mathématique de la gestion de l'autonomie est celui de la planification stratégique en deux phases et nous ne répétons pas ce modèle.

#### 4.4.1. Modélisation en programme linéaire du problème de configurations de robots

Une fois que le plan des rechargements des robots est établie, on procède à la configuration des robots. Le but de cette étape est de minimiser les dépassements énergétiques par rapport aux recommandations de consommations  $E_n$  d'énergie déterminées précédemment.

Les variables de décisions à prendre en compte dans cette phase sont :

$$y_{nc} = \begin{cases} 1, & \text{si le robot } n \text{ est configuré selon la configuration } c \\ 0, & \text{sinon.} \end{cases}$$

$$z_{nm} = \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot } n \\ 0, & \text{sinon.} \end{cases}$$

Le problème de configuration du système robots peut être formulé comme suit :

$$\text{Minimize } \sum_{n \in N} s_n \quad (4-80)$$

sous contraintes:

$$\sum_{c \in C} y_{nc} = 1, \quad \forall n \in N \quad (4-81)$$

$$\sum_{n \in N} z_{nm} = 1, \quad \forall m \in M \quad (4-82)$$

$$\sum_{c \in C} \sum_{n \in N} a_{fc} y_{nc} \leq N_f, \quad \forall f \in F \quad (4-83)$$

$$\sum_{m \in M} \phi_m e_m z_{nm} \leq s_n + E_n, \quad \forall n \in N \quad (4-84)$$

$$\sum_{c \in C_m} y_{nc} \geq z_{nm}, \quad \forall m \in M, \forall n \in N \quad (4-85)$$

$$s_n + E_n \leq E_{n,\max} \quad \forall n \in N \quad (4-86)$$

$$y_{nc}, z_{nm} \in \{0,1\} \text{ and } s_n \geq 0 \quad \forall b \in B; m \in M; n \in N \quad (4-87)$$

où  $s_n$  représente le dépassement énergétique par rapport aux recommandations établies dans la phase précédente de gestion d'autonomie. Les contraintes (4.81), (4.82), (4.83), (4.85), (4.86), (4.87) sont identiques à ceux du problème combiné de localisation et de configuration dans la planification stratégique en deux phases. Les contraintes (4.84) déterminent le dépassement énergétique par rapport à la recommandation de consommation d'énergie  $E_n$  du robot  $n$ .

Dans le cas où la gestion d'autonomie n'est pas nécessaire, on peut prévoir une autre formulation qui a pour but d'équilibrer les charges de travail ou bien de minimiser la charge maximale de travail.

$$\text{Minimize } \delta \quad (4-88)$$

sous contraintes:

$$\sum_{c \in C} y_{nc} = 1, \quad \forall n \in N \quad (4-89)$$

$$\sum_{n \in N} z_{nm} = 1, \quad \forall m \in M \quad (4-90)$$

$$\sum_{c \in C} \sum_{n \in N} a_{fc} y_{nc} \leq N_f, \quad \forall f \in F \quad (4-91)$$

$$\sum_{m \in M} T_m z_{nm} \leq \delta, \quad \forall n \in N \quad (4-92)$$

$$\sum_{c \in C_m} y_{nc} \geq z_{nm}, \quad \forall m \in M, \forall n \in N \quad (4-93)$$

$$y_{nc}, z_{nm} \in \{0,1\} \text{ and } \delta, \delta' \geq 0 \quad \forall b \in B; m \in M; n \in N \quad (4-94)$$

Les contraintes (4.89), (4.90), (4.91), (4.93), (4.94) sont identiques aux les contraintes présentés dans la formulation. La fonction objectif (4.88) a pour but d'équilibrer la charge de travail des robots. Les contraintes (4.92) définissent la charge de travail minimal et maximal. On note que seulement les temps d'exécution des missions sont pris en compte dans le calcul des charges de travail.

#### 4.4.2. Modélisation en programme linéaire du problème de localisation des stations

Le but de cette phase est de minimiser les temps de parcours en tenant compte les recommandations énergétiques des phases précédentes. Les notations et variables utilisées pour formuler les problèmes combinés de localisation et configurations sont toujours valables.

Les variables de décision sont :

$$x_{nh} = \begin{cases} 1, & \text{si le robot } n \text{ est assigné à la station d'attente } h \\ 0, & \text{sinon.} \end{cases}$$

$$z_{nm} = \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot } n \\ 0, & \text{sinon.} \end{cases}$$

$$u_m = \begin{cases} 0, & \text{si la mission } m \text{ est prise en charge par un robot.} \\ 1, & \text{sinon.} \end{cases}$$

Le produit  $x_{nh}z_{nm}$  est représenté par la variable de décision :

$$r_{nhm} = \begin{cases} 1, & \text{si la mission } m \text{ est affectée au robot } n \text{ et dont la station d'attente est } h \\ 0, & \text{sinon.} \end{cases}$$

Le problème de localisation des stations d'attentes est formulé comme suit:

$$\text{Minimize } \sum_{n \in N} \sum_{h \in H} \sum_{m \in M} \phi_m (A_{hm} + T_m + R_{hm}) r_{nhm} + \sum_{n \in N} \chi s_n - \sum_{m \in M} p_m u_m \quad (4-95)$$

sous contraintes:

$$\sum_{h \in H} x_{nh} = 1, \quad \forall n \in N \quad (4-96)$$

$$\sum_{n \in N} z_{nm} + u_m = 1, \quad \forall m \in M \quad (4-97)$$

$$\sum_{n \in N} x_{nh} \leq H_h, \quad \forall h \in H \quad (4-98)$$

$$\sum_{h \in H} \sum_{m \in M} \phi_m (A_{hm} + T_m + R_{hm}) r_{nhm} \leq T_{\max}, \quad \forall n \in N \quad (4-99)$$

$$\sum_{m \in M} \phi_m e_m z_{nm} \leq s_n + E_n, \quad \forall n \in N \quad (4-100)$$

$$\sum_{m \in M} \phi_m e_m z_{nm} \leq E_{n, \max}, \quad \forall n \in N \quad (4-101)$$

$$r_{nhm} \geq (x_{nh} + z_{nm}) - 1, \quad \forall m \in M, n \in N, h \in H \quad (4-102)$$

$$\sum_{c \in C_m} Y_{nc} \geq z_{nm}, \quad \forall m \in M, \forall n \in N \quad (4-103)$$

$$x_{nb}, z_{nt}, r_{nbt} \in \{0, 1\}, \quad \forall b \in B, \forall t \in T, \forall n \in N \quad (4-104)$$

où  $\gamma$  est une variable de décision réelle qui représente la violation maximale des recommandations énergétiques.

Les contraintes (4.95), (4.96), (4.97), (4.98), (4.99), (4.100), (4.102), (4.104) sont identiques aux contraintes (4.40), (4.41), (4.43), (4.45), (4.46), (4.49), (4.48), (4.51) du problème combiné de configuration et de localisation. Les contraintes (4.101) assurent que la quantité d'énergie requise pour exécuter les missions affectées ne dépasse la quantité maximale stockée sur les batteries. Les contraintes (4.103) garantissent que les missions sont affectées aux robots ayant une configuration adéquate fixée lors de l'étape précédente.

## **4.5. Comparaison et expérimentations numériques**

### **4.5.1. Exemples illustratifs**

Afin de mieux cerner les différences entre les différentes approches et ainsi que les différentes phases de résolution, nous présenterons deux exemples. Le premier est de petite taille, ce qui permet de résoudre à l'optimalité l'approche combinée et nous nous intéressons dans cet exemple à la charge du travail des robots dans la première période après la résolution de toutes les phases de l'approche donnée. Dans le deuxième exemple, nous considérons une instance de taille moyenne, et nous nous intéressons au changement de la charge de travail à travers les différentes phases de résolutions en utilisant l'approche de trois phases. Nous utilisons la loi uniforme pour générer les durées d'exécution de mission ainsi que les points de départ et d'arrivée qui sont générés aléatoirement à partir d'un plan donné. La vitesse des robots est fixée à 0.5m/sec et la durée d'un trajet est calculée à partir de cette vitesse.

#### **Exemple 1 : Comparaison entre les trois approches.**

Afin de comparer l'effet des trois approches (intégrée / combinée, 2 phases et 3 phases) sur la charge du travail des robots, nous considérons un système composé de 4 robots ( $N=4$ ), disposant de 7 stations d'attente disponibles ( $H=7$ ). Chaque robot dispose de 9 configurations par l'ajout de fonctionnalités ( $|C|=9$ ). Ces robots doivent exécuter un ensemble de 15 missions ( $|M|=15$ ) sur chacun de deux créneaux ( $T=2$ ) considérés pour l'optimisation. Chaque station d'attente peut accueillir un robot.

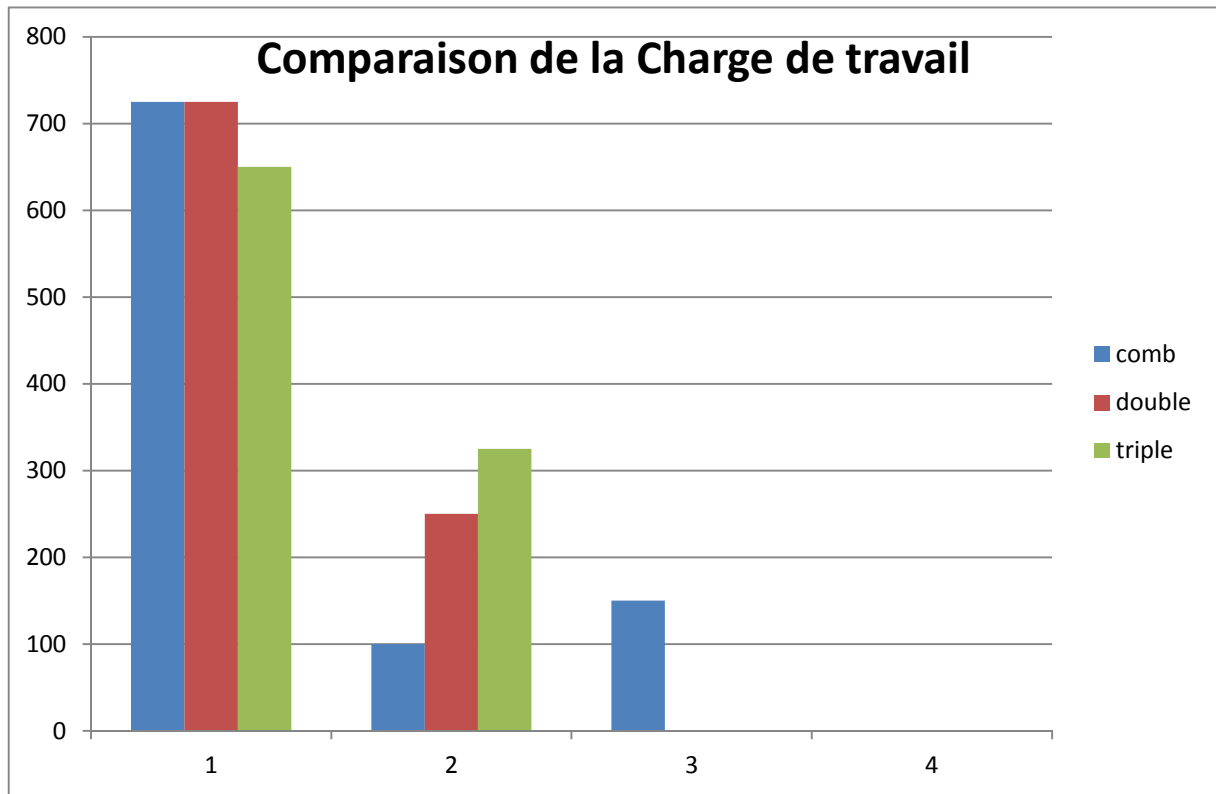


Figure 20 : Charge de travail de chaque robot

La Figure 20 illustrant la charge de travail de chaque robot, montre que l'approche combinée, privilégie la réactivité du système en utilisant un système de 3 robots, tandis que dans les deux autres approches seulement deux robots sont opérationnels, puis ce qu'en première phase de gestion d'autonomie deux robots sont amenés à se recharger, et ce qui entraîne que la charge de travail doit être répartie sur ces deux robots disponibles.

### Exemple 2 : Charge de travail des robots dans une résolution en trois phases.

Afin d'étudier l'impact de la résolution en trois phases sur la charge de travail, nous considérons un système robotique composé de 8 robots ( $N=8$ ) pouvant être configurés de 9 façons différentes ( $|C|=9$ ) et disposant de 12 stations d'attente ( $|H|=12$ ). Ces robots doivent exécuter 20 missions ( $|M|=20$ ) sur chacun des deux créneaux considérés ( $T=2$ ).

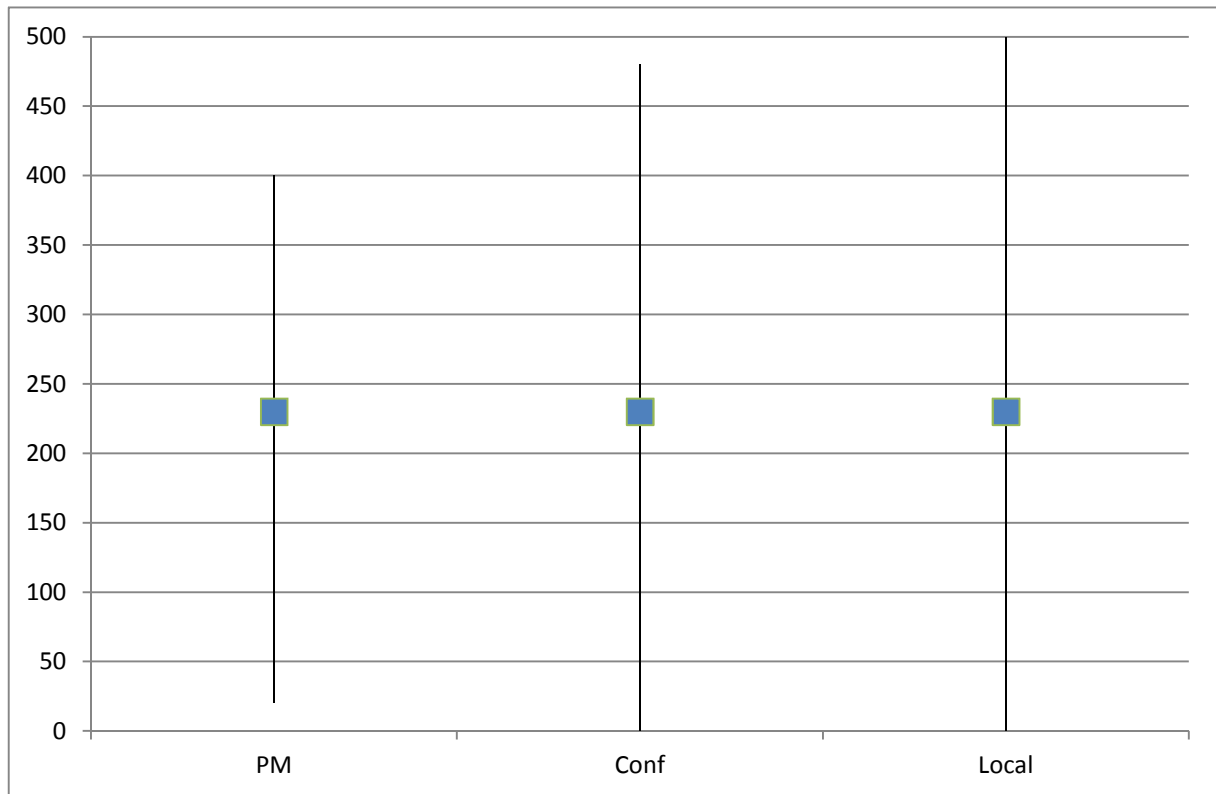


Figure 21 : Charge de travail dans les 3 phases

La Figure 21, illustrant la charge moyenne par robot ainsi que la charge maximale et minimale sur chacune des trois phases, montre qu'en utilisant la résolution MIP en 3 phases, la charge de travail est mieux équilibrée en première phase (PM) entre les différents robots actifs. Ensuite en deuxième phase (Conf) lors de l'attribution des configurations, la charge de travail de certains robots est augmentée et diminuée pour les autres. Le même phénomène est amplifié lors de l'attribution de station d'attente en troisième phase (Local).

#### 4.5.2. Comparaison numériques entre les différentes approches de résolution MIP

Afin de tester la complexité de chaque approche ainsi que la qualité des résultats fournis, nous testons cinq instances présentées dans le tableau suivant

Instance	N	H	T	M	C
1	4	7	2	15	9
2	4	8	2	14	9
3	5	9	4	17	9
4	4	7	2	15	9
5	15	20	2	40	9

Tableau 6: Caractéristiques des Instances

Où ( $|N|$ ) représente les nombre de robots, ( $|H|$ ) représente le nombre de stations d'attente considérées. On note qu'on limite la capacité de chaque station à un seul robot. ( $|M|$ ) est le nombre de missions à exécuter, ( $|C|$ ) est le nombre de configurations possibles et finalement ( $|T|$ ) représente le nombre de créneaux ou de périodes sur lesquels le problème est considéré.

Bien évidemment en utilisant l'approche intégrée ou combinée, la résolution se fait en une seule phase. Quant à la résolution en deux phases, en première phase le problème énergétique est seulement considéré, ensuite en deuxième phase le problème de configuration est combiné avec le problème de localisation et ils sont résolus ensemble en en tenant compte des recommandations de la première phase. Afin de comparer la qualité des résultats, nous considérons une deuxième fonction objectif. Cette fonction objectif est calculée, une fois que le problème initial est résolu, et seule la charge effective de travail incluant les durées des trajets d'aller et de retour est considérée.

### Instance 1

Méthode	CPU 1 <sup>ère</sup> phase	CPU 2 <sup>ème</sup> phase	CPU 3 <sup>ème</sup> phase	Fonc. Obj
MIP combiné	44			3451
MIP en 2 phases	1<	1<		3603
MIP en 3	1<	1<	1<	3557

### Instance 2

Méthode	CPU 1 <sup>ère</sup> phase	CPU 2 <sup>ème</sup> phase	CPU 3 <sup>ème</sup> phase	Fonc. Obj
MIP combiné	7			3450
MIP en 2 phases	1<	1<		3450
MIP en 3 phases	1<	1<	Pas de solution	N /A

### Instance 3

Méthode	CPU 1 <sup>ère</sup> phase	CPU 2 <sup>ème</sup> phase	CPU 3 <sup>ème</sup> phase	Fonc. Obj
MIP combiné	2709>			N /A
MIP en 2 phases	1<	1		453.75
MIP en 3 phases	1<	1<	1<	453.75

### Instance 4

Méthode	CPU 1 <sup>ère</sup> phase	CPU 2 <sup>ème</sup> phase	CPU 3 <sup>ème</sup> phase	Fonc. Obj
MIP combiné	44			3451
MIP en 2 phases	1<	1<		3603
MIP en 3 phases	1<	1<	1<	3557

### Instance 5

Méthode	CPU 1 <sup>ère</sup> phase	CPU 2 <sup>ème</sup> phase	CPU 3 <sup>ème</sup> phase	Fonc. Obj
MIP combiné	Time Out			N/A
MIP en 2 phases	1<	2746		3603
MIP en 3 phases	1<	1<	684	N/A

Ces expérimentations ont été exécutées sur un nœud de cluster linux à 8 processeurs Intel Xeon 64 bits avec une mémoire partagées de 8GB. Le solveur Cplex de IBM a été utilisé comme solveur MIP avec une Licence, permettant l'exécution en parallèle sur tous les processeurs.

Ces résultats montrent que la complexité de la méthode combinée la rend inexploitable même pour les petites instances, comme le montre les instances 3 et 5 qui sont relativement des instances de petite tailles mais et que la méthode combiné n'a pas pu résoudre ces problèmes. Tandis que la résolution en deux ou trois phases peut être exécutée en moins d'une seconde pour ces instances de petites tailles. En revanche l'instance 2, met en évidence, un des inconvénients de la méthode de trois phases, où la configuration des robots en deuxième phase, rétréci l'espace des solutions faisables en troisième phase et dans notre exemple aucune solution faisable na pas pu être trouvé.

Et les expérimentations montrent que la résolution du problème de localisation est un peu plus complexe que les autres problèmes traités. Une nouvelle modélisation compacte du problème joint (localisation - configuration) a confirmé ces résultats. Cette nouvelle modélisation permet de résoudre certaines instances en deux phases plus rapidement qu'en trois phases en utilisant l'ancien modèle.

### 4.5.3. Expérimentations numériques utilisant la méthode de générations de colonnes

Dans cette section, nous considérons que les robots sont équipés de batteries amovibles et le problème de gestion d'autonomie n'est pas considéré. Donc seulement la résolution des deux problèmes combinés : la configuration et la localisation ; en utilisant la méthode de génération de colonnes est considérée dans cette section. Nous proposons de tester les performances de cette méthode, sur les cinq instances présentées dans le tableau suivant :

Instance	N	H	C	M	Tmax
1	2	3	3	7	100 min
2	3	7	6	10	100 min
3	4	7	6	20	120 min
4	5	14	12	30	120 min
5	6	14	12	40	120 min

Tableau 7: Caractéristiques des instances de générations de colonnes

Où ( $|N|, |H|, |C|$ ) et  $|M|$ ) représentent respectivement le nombre de robots, de stations d'attente, le nombre de configurations possibles et le nombre de missions à exécuter.

Ces expérimentations ont été exécutées sur une machine Windows équipée d'un processeur 2Ghz avec 3 Gb de Ram. Le problème maitre est résolu en utilisant le solveur open source CLP de la librairie COIN-OR. Le tableau suivant présente les performances de la méthode de générations de colonnes appliquées sur notre problème

Instance	Fonction Objective	# Col.	# Itérations	#Proced. Arrond.	Gap %	Temps. (sec)
1	1830	26	10	0	0.0	<1
2	2030	157	8	0	0.0	7
3	4090	544	36	1	0.18	74
4	5980	1473	26	0	0.0	435
5	8290	3459	112	4	0.03	889

Tableau 8: Performances de la méthode de générations de colonnes



Où (#col.) représente le nombre de colonnes générées, (# Itérations) est le nombre d'itérations entre problème maître et esclave avant d'atteindre le critère d'arrêt. (#Proced. Arrond.) représente le nombre de fois que l'on a eu recours à l'heuristique d'arrondissement des colonnes fractionnelles afin d'atteindre une solution en nombres entiers.

On note que pour les instances de petite taille ( $\#N < 4$  et  $\#M < 20$ ), le modèle compact MIP est plus performant que la génération de colonnes, toutefois cette dernière surpasse le MIP pour les instances de taille moyenne ( $\#N = 6$  et  $\#M = 40$ ). Par exemple la résolution utilisant la génération de colonnes nécessite 889 secondes de temps de calcul. Cependant la résolution MIP en utilisant CPLEX nécessite plus que 4 jours de temps de calcul sur le cluster.

Sachant que 70% du temps de la résolution est consacré au problème esclave, une nouvelle version permettant de paralléliser l'exécution du problème esclave sur le cluster est dans les dernières phases de développement.

Nos collègues de Cardiff ont repris notre modèle joint de localisation et configuration proposé dans [Baalbaki 08 a] et ils ont proposé de le résoudre en utilisant des méta-heuristiques inspirées du comportement des abeilles [Xu 10]. Nous avons pris contact avec eux afin de comparer les performances de nos méthodes.

Un site Web consacré pour ce problème a été mis en place afin de fournir aux chercheurs, une introduction de ce problème, les instances sur lesquels nous avons travaillé ainsi que la meilleure solution trouvée par approche et par instances.