

Formalizzazione del Task Assignment Problem

20 dicembre 2009

1 Definizione del problema

E' necessario pervenire ad una modellizzazione del problema del task assignment che permetta di adottare tecniche note per l'identificazione delle variabili decisionali, dei vincoli e della funzione obiettivo da massimizzare. Ciò è fondamentale anche per lo studio matematico delle prestazioni. A tal fine si introduce un modello di programmazione lineare, dapprima considerando ipotesi molto semplici.

1.1 Primo modello

L'approccio è di tipo statico e centralizzato, per il momento. Si considera una precisa configurazione del pool dei task e si cerca di ottimizzare l'assegnazione. Si assume nota la matrice di copertura $V \in [0, 1]^{N \times M}$, dove N è il numero degli agenti ed M quello delle aree visive. Si considera che ad ogni task sia associata una struttura che ne riporta la priorità e l'area visiva di pertinenza (inizialmente la logica adottata è di tipo *hard logic*). Il modello proposto è:

$$x_{i,j} = \begin{cases} 1 & \text{se l'agente } i \text{ svolge il task } j \\ 0 & \text{altrimenti} \end{cases}$$

$$\bar{v}_{i,j} = \begin{cases} 1 & \text{se l'agente } i \text{ non copre l'area } task(j).loc \\ 0 & \text{altrimenti} \end{cases}$$

$$\sum_{j=1}^K x_{i,j} \leq 1 \quad \forall i \quad \text{ogni agente svolge al massimo un task} \quad (1)$$

$$\sum_{i=1}^N x_{i,j} \leq 1 \quad \forall j \quad \text{ogni task è assegnato al max ad un agente} \quad (2)$$

$$\sum_{j=1}^K \bar{v}_{i,j} x_{i,j} = 0 \quad \forall i \quad \text{gestione delle aree di copertura ottimizzata} \quad (3)$$

$$x_{i,j} \in \{0, 1\} \quad \forall i, j$$

$$\max \sum_{i,j} p_{i,j} x_{i,j}$$

Si noti come la natura dei vincoli 3 non permetta di concludere sulla totale unimodularità della matrice di vincolo.

Questa modellizzazione lascia irrisolti alcuni problemi. In primis, il criterio di scelta dei task sulla base della sola priorità potrebbe introdurre troppa rigidità, pervenendo sistematicamente a scartare i task a priorità inferiore (streaming e patrolling). Si potrebbe pensare ad introdurre un criterio di tipo probabilistico, oppure a pesare la priorità di un task con il tempo di permanenza nel pool.

Il vincolo 2, inoltre, impedisce che un task possa essere assegnato a più agenti: si potrebbe rilassarlo, ad esempio nel caso in cui il task in questione sia di tipo patrolling. Una richiesta ulteriore, non ancora modellata a questo punto, è infatti che ci sia continuità nel monitoraggio di ciascuna area visiva. Semplificando il problema, dal momento che il task di streaming è incompatibile, nelle nostre ipotesi, con la registrazione di immagini attuali, ciò che si vorrebbe imporre è che per ogni area almeno una camera esegua brandeggio manuale, tracking oppure patrolling (in modo da registrare in tempo reale quanto avviene nell'area di pertinenza). Non si considera ancora che il movimento delle camere ne modifichi l'area di copertura (si assume in pratica che per ciascun agente l'area visiva sia l'unione di tutte quelle visibili nelle diverse configurazioni). Il nuovo vincolo può essere introdotto in modo esplicito, ad esempio introducendo le variabili:

$$m_{ijh} = \begin{cases} 1 & \text{se l'agente } i \text{ è tale che l'area } h \in V(i, :) \text{ e } task(j).type \in [o, t, p] \\ 0 & \text{altrimenti} \end{cases}$$

e le disequazioni

$$\sum_{i=1}^N \sum_{j=1}^K m_{ijh} x_{ij} \geq 0 \quad \forall h \in \{1, \dots, M\} \quad (4)$$

Si ricorda che ad ogni task j del pool sono associate l'area di pertinenza $task(j).loc$ e la tipologia $task(j).type$ che può essere *brandeggio operatore*, *o,tracking*, *t,streaming*, *s*, *patrolling*, *p*.

Un'alternativa, utile a semplificare il problema, potrebbe essere modellare la richiesta di continuità del monitoraggio non tramite i vincoli 4, ma attraverso la funzione obiettivo. Con significato delle variabili analogo al precedente, si riscrive il modello con vincoli 1, 2, 3 e si considera:

$$\max \sum_{i,j} p_j x_{ij} + \sum_h \sum_{i,j} m_{ijh} x_{ij}$$

2 Secondo modello

Un approccio differente potrebbe essere quello di dissociare il problema considerandone separatamente le due fasi di scelta dei task da assegnare, tra quelli sussistenti, ed effettiva assegnazione dei task selezionati agli agenti. Si potrebbe modellare il primo problema (*Task Selection*) tramite un modello di programmazione lineare simile al Multiple Knapsack Problem. Si considerano M classi p_m di task, una per area, con l_m elementi coincidenti con i task, e si associa a ciascun'area una capacità massima c_i pari, per esempio, al numero di agenti di

quell'area. Si può scrivere un modello del tipo:

$$\begin{aligned}
 x_{im} &= \begin{cases} 1 & \text{se il task } i \text{ di area } m \text{ viene selezionato} \\ 0 & \text{altrimenti} \end{cases} \\
 \max & \sum_{m=1}^M \sum_{i=1}^{l_m} g_{im} x_{im} \\
 \sum_{i=1}^{l_m} a_i x_{im} &\leq c_m \quad \forall m \\
 x_{im} &\in \{0, 1\}
 \end{aligned} \tag{5}$$

Nel quale g_{im} è il guadagno associato alla selezione del task x_{im} ed a_{im} è la richiesta di risorse implicata (se pari ad 1 significa che il task richiede di occupare un agente). La seconda fase (che potremmo chiamare *Task Deployment*), consiste nel distribuire i task tra gli agenti e potrebbe essere affrontata utilizzando logiche tipo market-based, assegnazione fissa oppure randomizzata.

Si tratta quindi di una formulazione piuttosto generale, che ha il pregio di separare i due momenti del problema, permettendo ad esempio una gestione ibrida centralizzata-distribuita per la prima e la seconda fase, rispettivamente. Questo modello potrebbe essere sviluppato ulteriormente, quindi.

Un'altra questione da affrontare è riscrivere i modelli di PL in modo tale da semplificarne la risoluzione e decidere quali tecniche utilizzare per ottenere soluzioni. Una buona idea potrebbe essere indagare le varianti del problema KP.