

Relazione progetto

Reti di calcolatori

Indice

Componenti del gruppo	1
Descrizione delle specifiche	2
Algoritmo del matrimonio stabile	2
Prima generalizzazione: Insiemi di cardinalità differente	3
Seconda generalizzazione: Liste di preferenza incomplete	3
Implementazione	4
Struttura del codice	4
Rappresentazione della rete	4
Assegnazione dei dispositivi	4
Rappresentazione grafica	5
Bibliografia	5

Componenti del gruppo

Nome Cognome	Matricola	Email
Leonardo Calbi	6155786	leonardo.calbi@stud.unifi.it
Alessio Falai	6134275	alessio.falai@stud.unifi.it

Anno accademico 2018/2019, CdL Informatica (Scuola SMFN).

Descrizione delle specifiche

Lo scopo del progetto è quello di simulare una rete di telecomunicazioni, composta da un insieme di nodi e un insieme di dispositivi. Ogni dispositivo deve essere associato ad un unico nodo affinché esso possa computare l'operazione richiesta. Ogni nodo può accettare un numero prefissato di dispositivi.

Per rispettare le richieste si presuppone che il grafo di rete abbia un'unica componente connessa, in modo da garantire la raggiungibilità di ogni nodo da parte di ciascun dispositivo.

Ciascun dispositivo ha al suo interno una lista di preferenza relativa ai nodi e ciascun nodo ha una lista di preferenza implicita relativa ai dispositivi, sulla base della loro priorità.

Lo scopo del progetto è dunque quello di associare nodi e dispositivi utilizzando un'implementazione dell'algoritmo del matrimonio stabile con liste di preferenza incomplete.

Algoritmo del matrimonio stabile

Consideriamo due insiemi composti da n elementi ciascuno: $X = \{x_1, \dots, x_n\}$ e $Y = \{y_1, \dots, y_n\}$. Un matching M tra i due insiemi X e Y è un sottoinsieme del prodotto cartesiano $X \times Y$, costituito da una famiglia di n coppie tale che ciascun elemento di X e ciascun elemento di Y appartengano ad una ed una sola coppia. Ad esempio, se $X = \{a, b, c\}$ e $Y = \{\alpha, \beta, \gamma\}$, un matching tra X e Y è $M = \{(a, \beta), (b, \gamma), (c, \alpha)\}$. Associamo ad ogni elemento $x_k \in X$ una n -pla ordinata costituita da una permutazione di tutti gli elementi di Y : $P(x_k) = (y_{k_1}, \dots, y_{k_n})$, dove con (k_1, k_2, \dots, k_n) abbiamo indicato una permutazione degli indici $(1, 2, \dots, n)$. Analogamente, associamo a ciascun elemento $y_h \in Y$ una n -pla ordinata costituita da una permutazione di tutti gli elementi di X : $P(y_h) = (x_{h_1}, \dots, x_{h_n})$. Per ogni elemento $x_k \in X$ la permutazione $P(x_k)$ rappresenta una lista di preferenza degli elementi di Y per x_k ; al tempo stesso la permutazione $P(y_h)$ rappresenta l'ordine di gradimento degli elementi di X per ciascun $y_h \in Y$. In altri termini ciascun elemento di X esprime una propria classifica sugli elementi di Y disponendoli in ordine decrescente di gradimento; lo stesso fanno gli elementi di Y esprimendo le proprie preferenze sugli elementi di X . Se $a \in X$, $\alpha, \beta \in Y$ e se a preferisce α a β , scriveremo $\alpha <_a \beta$, indicando così con il simbolo " $<_a$ " la relazione d'ordine totale definita da a sugli elementi di Y .

Una volta introdotte le "liste di preferenza" degli elementi di un insieme rispetto agli elementi dell'altro insieme, tra le coppie di un matching può venirsi a creare una condizione di forte instabilità. Sia M un matching tra X e Y e siano $(a, \alpha), (b, \beta) \in M$ due coppie del matching tali che β precede α nella lista di preferenza $P(a)$ di a e, al tempo stesso, a precede b nella lista di preferenza $P(\beta)$ di β ; in simboli, utilizzando la notazione appena descritta, possiamo scrivere $\beta <_a \alpha$ e $a <_\beta b$. Allora a e β avrebbero l'interesse reciproco di "tradire" i propri compagni e mettere in crisi i propri "matrimoni" costituendo la nuova coppia (a, β) in cui ciascuno dei due membri, a e β , migliorerebbero la propria condizione.

Il problema del matrimonio stabile mira a definire un matching in cui una simile condizione di instabilità non esista per nessuna coppia di elementi; in altri termini, dati i due insiemi X e Y si vuole costruire un matching tra membri dei due insiemi in modo tale da non creare una situazione di instabilità che, se il matching fosse un insieme di matrimoni, ne metterebbe presto in crisi qualcuno. Un matching si dice stabile se la condizione di instabilità non si verifica tra le coppie del matching.

Prima generalizzazione: Insiemi di cardinalità differente

Supponiamo che la cardinalità dei due insiemi non sia la stessa, ad esempio $|X| < |Y|$. In questa condizione non può essere definita un'istanza del problema matrimonio stabile, perché qualche elemento di Y rimarrebbe comunque privo di compagno, a meno di non consentire ad uno stesso elemento di X di essere associato a più di un elemento dell'insieme Y . Se $|X| < |Y|$, con l'ipotesi aggiuntiva che ogni elemento $x_k \in X$ possa essere associato a più di un elemento di Y , il problema può essere ricondotto al problema del matrimonio stabile con insiemi di uguale cardinalità, costruendo un'estensione di X , ripetendo n_k volte ciascun elemento $x_k \in X$, fino a costituire un insieme X' con la stessa cardinalità di Y .

Seconda generalizzazione: Liste di preferenza incomplete

Supponiamo quindi che a ciascun elemento $x_k \in X$ del primo insieme sia associata una lista di preferenze parziale $P(x_k)$ sui membri dell'altro insieme e che lo stesso avvenga per ogni elemento $y_h \in Y$ del secondo insieme.

L'obiettivo è quello di ricondurre ogni istanza del problema con liste di preferenza incomplete ad una istanza equivalente del problema del matrimonio stabile con liste di preferenza complete.

Per far questo aggiungiamo ad ognuno dei due insiemi un elemento, che chiameremo il *vedovo* e la *vedova*: $X = X \cup \{v_X\}$, $Y = Y \cup \{v_Y\}$. Le liste di preferenza $P(x_k)$ e $P(y_h)$ degli elementi di X e di Y vengono completate come segue, in modo tale da ricondursi al caso del problema con liste di preferenza complete:

- si colloca v_X come ultima preferenza di v_Y e viceversa
- si colloca v_X come ultima preferenza di ciascun elemento $y \in Y$, e viceversa
- si aggiungono in un ordine arbitrario, dopo v_X tutti gli elementi mancanti alla lista di ciascun elemento $y \in Y$, e viceversa

Implementazione

Struttura del codice

Il codice è suddiviso nelle seguenti classi:

- Node: Rappresenta un nodo del grafo connesso, identificato da un determinato ID. Per allocare un nodo è necessario conoscere il massimo numero di dispositivi a esso associabili. Le strutture presenti all'interno di ciascuna istanza sono l'insieme dei nodi vicini e la coda con priorità per la gestione delle richieste di accesso dei dispositivi.
- Priority: Tipo enumerativo per identificare un ordine tra le 5 classi di priorità dei dispositivi.
- Device: Rappresenta un dispositivo di rete, identificato da un determinato ID e da una priorità. Inizialmente ogni device viene considerato connesso a un nodo della rete. Per richiedere l'associazione con un nodo delega la ricerca alla classe di esplorazione Probe.
- Probe: Esplora il grafo a partire da un nodo, tramite due strutture dati, una frontiera e un insieme di nodi già visitati.

Rappresentazione della rete

La rete di telecomunicazioni è stata implementata mediante una struttura dati di tipo grafo, in cui ciascun nodo contiene l'insieme che identifica i propri vicini.

Si presuppone che ciascun dispositivo sia inizialmente collegato a un nodo, scelto in modo casuale. Il numero di nodi, archi e dispositivi viene scelto arbitrariamente all'avvio del programma, mentre il numero di dispositivi associabili a ciascun nodo viene selezionato effettuando una partizione del numero totale U di dispositivi di rete, in modo tale che le risorse dei nodi sommino a U .

Il grafo è stato costruito in modo tale da garantirne la connessione. Per fare ciò è necessario che il numero di archi sia almeno pari al numero dei nodi meno uno e che ogni "nuovo" nodo venga collegato a un nodo che abbia già almeno un altro arco. In altre parole, inizialmente puntiamo alla creazione di un albero. Per generare il grafo, altri archi vengono poi aggiunti randomicamente.

Assegnazione dei dispositivi

L'assegnazione dei dispositivi ai nodi viene effettuata utilizzando una variante dell'algoritmo del matrimonio stabile con liste incomplete, sopra descritto. In questa variante le preferenze espresse da ciascun dispositivo dipendono esclusivamente dalla distanza col nodo di computazione, rispetto al nodo a cui si trovano inizialmente collegati i dispositivi. Per quanto riguarda i nodi, la loro lista di preferenza nei confronti dei dispositivi può essere espressa mediante un ordinamento sulla base del loro valore di priorità. L'assegnazione prevede lo svolgimento delle seguenti fasi:

1. Il dispositivo tenta di accedere alle risorse del nodo scelto, che inizialmente equivale al nodo a lui direttamente connesso

2. Nel caso in cui il nodo scelto possa ospitare il dispositivo in questione, allora viene effettuata l'allocazione e la ricerca termina. Altrimenti, nel caso in cui la lista di risorse del nodo sia piena e la priorità del dispositivo entrante sia minore della minima priorità presente nell'insieme di dispositivi allocati al nodo, il dispositivo continua l'esplorazione del grafo. Se invece, la priorità del dispositivo entrante è maggiore della minima priorità presente nell'insieme di dispositivi allocati al nodo, il dispositivo corrente viene allocato, mentre quello con priorità minima riprende la ricerca di un nodo
3. Il nodo viene aggiunto a un insieme di nodi visitati, per evitare la creazione di cicli. In seguito, il dispositivo si occuperà di esplorare i nodi vicini, utilizzando una struttura di tipo frontiera del grafo, per mantenere le minime distanze dalla sorgente
4. Se tutti i nodi a distanza minima sono già stati visitati, si passa all'esplorazione dei nodi immediatamente successivi in termini di distanza, altrimenti si tenta di accedere alle risorse di un nodo scelto randomicamente dalla frontiera, a distanza minima
5. Le operazioni descritte vengono ripetute fino a che tutti i dispositivi non sono stati correttamente allocati

L'allocazione dei dispositivi avviene in modo parallelo, dato che l'esplorazione del grafo viene effettuata da un thread diverso per ciascun dispositivo.

Rappresentazione grafica

L'esecuzione del programma presenta un'interfaccia grafica con due diverse rappresentazioni della rete:

1. La prima figura mostra lo stato iniziale del grafo, in cui i nodi di computazione sono di colore azzurro, mentre i dispositivi sono di colore giallo. L'etichetta identificante ciascun nodo contiene il nome del nodo e il numero di dispositivi a esso associabili, mentre l'etichetta identificante ciascun dispositivo contiene il nome del dispositivo e la sua priorità, in una scala che va da 1 (minima) a 5 (massima). Il grafo in questione presenta dunque i collegamenti iniziali fra dispositivi e nodi.
2. La seconda figura mostra lo stato finale del grafo, in cui ogni dispositivo è collegato al nodo a cui esso è associato. I colori utilizzati sono gli stessi del grafo di partenza, in modo da poterli agevolmente confrontare.

Bibliografia

- ❖ [Il problema del Matrimonio Stabile](#) (Marco Liverani)