



METODI ED ALGORITMI DI OTTIMIZZAZIONE PER IL PROBLEM SOLVING

Docente:
Aristide Mingozzi

Adattamento:
Edoardo Rosa

Anno Accademico 20014-2015

CONTENTS

1	Modelli e formulazioni matematiche	1
1.1	The Traveling Salesman Problem	1
1.2	Formulazioni Matematiche del TSP	1
1.2.1	TSP asimmetrico	2
1.2.2	TSP simmetrico	2
1.2.3	Eliminazione subtours di Miller, Tucker, Zemlin (1960)	3
2	Introduzione alla programmazione lineare a numeri interi	4
2.1	Stato attuale delle wireless network	4
2.2	Dinamic Power Management	4
A	Prova	6
A.1	Pippo	6
A.2	CiuCiu	8

LIST OF FIGURES

2.1	Consumo di energia senza power management	5
2.2	Consumo di energia con power management	5
2.3	Visione beacon e Ps-Poll	5

LIST OF TABLES

CHAPTER 1

MODELLI E FORMULAZIONI MATEMATICHE

1.1 The Traveling Salesman Problem

Il Traveling Salesman Problem (TSP) è il problema più noto dell'ottimizzazione combinatoria. Siano date n città e i costi c_{ij} per andare dalla città i alla città j . Si vuole determinare un cammino che parte da una città (diciamo i_1), visitare una ed una sola volta tutte le rimanenti città e terminare nella città di partenza i_1 . Inoltre si vuole che il costo di tale cammino sia minimo.

Ha molteplici applicazioni pratiche e teoriche perché è la struttura di molti problemi pratici. Si è soliti modellare il TSP come segue:

- è dato un grafo orientato (o non orientato) $G = (N, A)$ dove N è un insieme di n vertici e A è un insieme di m archi.

Ad ogni arco $(i, j) \in A$ è associato un costo c_{ij} .

Un circuito hamiltoniano di G è un circuito che passa per ogni vertice una ed una sola volta.

Il costo di un circuito hamiltoniano di G è pari alla somma dei costi degli archi che compongono il circuito;

- il problema del TSP è di trovare un grafo G , con una data matrice dei costi $[c_{ij}]$, un circuito hamiltoniano di costo minimo.

1.2 Formulazioni Matematiche del TSP

In letteratura esistono molteplici (e a volte fantasiose) formulazioni del TSP.

Presentiamo le due formulazioni più note e su cui si basano i metodi esatti più efficienti.

1.2.1 TSP asimmetrico

I costi c_{ij} non verificano $c_{ij} = c_{ji} \forall i, j$ con $i < j$.

Sia x_{ij} una variabile (0 – 1) associata ad ogni arco $(i, j) \in A$ dove $x_{ij} = 1$ se l'arco (i, j) è nella soluzione ottima e $x_{ij} = 0$ altrimenti.

$$\text{Min} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \quad (1.1)$$

$$\text{s.t.} \sum_{i \in N} x_{ij} = 1, \quad \forall j \in N \quad (1.2)$$

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in N \quad (1.3)$$

$$\sum_{i \in S} \sum_{j \in N \setminus S} x_{ij} \geq 1, \quad \forall S \subset N \quad (1.4)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \quad (1.5)$$

Il vincolo 1.4 impone che ogni soluzione ammissibile debba contenere almeno un arco (i, j) con $i \in S$ e $j \in N \setminus S$ per ogni sottoinsieme S di N . Un'alternativa al vincolo 1.4 è:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset N \quad (1.4')$$

1.2.2 TSP simmetrico

Sia dato un grafo non-orientato $G=(N, A)$ con $c_{ij} = c_{ji}, \forall i, j \in N$.

Gli archi di A sono numerati da 1 a m . L'arco di indice l corrisponde a (α_l, β_l) con $\alpha_l < \beta_l$.

A_i è il sottoinsieme degli indici degli archi che incidono sul vertice i :

$$A_i = \{l : l = 1, m \text{ s.t. } \alpha_l = i \text{ or } \beta_l = i\}$$

Per una dato $S \in N$ e $\bar{S} = N \setminus S$ indichiamo con (S, \bar{S}) il sottoinsieme degli indici degli archi per cui $\alpha_l \in S$ e $\beta_l \in \bar{S}$ oppure $\alpha_l \in \bar{S}$ e $\beta_l \in S$.

Ad ogni arco di indice l è associato un costo $d_l = c_{\alpha_l \beta_l}$ e $x_l \in \{0, 1\}$ è una variabile che vale 1 se e solo se l'arco di indice l è nella soluzione ottima.

$$\text{Min} \sum_{l=1}^m d_l x_l \quad (1.6)$$

$$\text{s.t.} \sum_{l \in A_i} x_l = 2, \quad \forall i \in N \quad (1.7)$$

$$\sum_{l \in (S, \bar{S})} x_l \geq 1, \quad \forall S \subset N \quad (1.8)$$

$$x_l \in \{0, 1\}, \quad l = 1, \dots, m \quad (1.9)$$

1.2.3 Eliminazione subtours di Miller, Tucker, Zemlin (1960)

Sia u_i una variabile intera il cui valore rappresenta la posizione che il vertice i occupa nel tour.

Es. tour (1,4,5,3,2,1) per TSP con $n=5$ vertici, si ha $u_1 = 1$, $u_2 = 5$, $u_3 = 4$, $u_4 = 2$, $u_5 = 3$

Miller, Tucker e Zemlin propongono in alternativa a:

$$\sum_{i \in S} \sum_{j \in N \setminus S} x_{ij} \geq 1, \quad \forall S \subset N \quad (*)$$

CHAPTER 2

INTRODUZIONE ALLA PROGRAMMAZIONE LINEARE A NUMERI INTERI

2.1 Stato attuale delle wireless network

Oggigiorno non esistono più dispositivi isolati che non comunicano con altri, tutti i PC prevedono la connessione ad internet, o in generale comunicano con altri dispositivi elettronici grazie alle reti di comunicazione. Inizialmente queste reti erano basate sui cavi (*wired network*), poi si sono rivelate più pratiche le reti senza filo (*wireless network*); da ciò si è dovuto affrontare il problema di permettere l'utilizzo delle tecnologie attualmente utilizzate nelle reti wired anche in reti wireless, che per loro natura sono intrinsecamente meno sicure a causa del mezzo trasmissivo. Le informazioni non vengono più convertite sotto forma di impulso elettronico e spedite su un filo metallico, ma essendo l'etere il mezzo di propagazione sono convertite in onde radio; quest'ultime hanno una portata limitata non sempre ben determinabile, sono soggette a disturbi che degradano le informazioni inviate fino, a volte, farle perdere; ed infine cosa a noi maggiormente sgradita l'energia consumata e direttamente connessa alla potenza trasmessa, ciò significa che aumentare la portata di trasmissione e la qualità del segnale comporta un consumo maggiore dell'energia consumata.

2.2 Dinamic Power Management

Essendo la parte a radio frequenza colpevole del maggior consumo di energia, in particolare gli amplificatori utilizzati immediatamente prima dell'invio del segnale o dopo una ricezione, le stazioni 802.11 possono allungare la vita delle batterie spegnendo i transceiver radio e mettendo la scheda wireless in sleep periodicamente.

Nella figura sottostante (Figura 2.1), si può osservare una finestra di 20(CTRLLLARE!!!) secondi del consumo di energia di una scheda wireless senza il power management abilitato:

Figure 2.1: Consumo di energia senza power management

Attivando il power management si può notare un sensibile risparmio energetico, a fronte di un diverso consumo energetico dovuto ad un diverso modo di operare: in questa modalità la

Figure 2.2: Consumo di energia con power management

stazione resta per la maggior parte del tempo nello stato di sleep, risvegliandosi periodicamente (nella figura ogni 100ms) per controllare se nel frattempo c'è stato traffico per lei. Durante il periodo di sleep, sarà demandato all'Access Point il compito di bufferizzare i frame destinati verso la scheda momentaneamente in sleep; al risveglio i frame bufferizzati saranno annunciati da un *Beacon frame*, la ricezione da parte della stazione wireless dei frame bufferizzati inizierà subito dopo con la loro richiesta tramite il *PS-Poll frame*. (Figura 2.3). Quando viene attivato

Figure 2.3: Visione beacon e Ps-Poll

il power management l'Access Point e la stazione wireless devono sincronizzarsi per risvegliarsi ad intervalli prestabiliti, che nel caso della figura soprastante è 100ms; all'inizio di ogni risveglio l'Access Point invierà il beacon contenente la *traffic indication map - TIM* che dice alla stazione se vi sono pacchetti bufferizzati destinati a lei; in tal caso la stazione risponderà con un PS-Poll contenente il suo Association ID. Grazie all'Association ID l'Access Point sarà in grado di selezionare i frame destinati alla stazione e provvederà ad inviarli.

APPENDIX A

PROVA

A.1 Pippo

A seguito della crescente diffusione di apparecchi portabili quali laptop, palmari, cellulari, ecc utilizzati in un numero sempre maggiore di luoghi pubblici e privati quali ospedali, aeroporti, uffici, aziende e negozi si sono dovute trovare soluzioni che conciliassero necessità diverse come interattività, molteplicità di funzioni, semplicità di utilizzo, maneggevolezza e riduzione degli ingombri.

I maggiori problemi riguardano il compromesso tra molteplicità di funzioni e riduzione degli ingombri, in quanto accessori aggiuntivi e funzioni complicate implicano risorse computazionali elevate a discapito di un maggiore consumo di energia. In particolar modo l'utilizzo di questi dispositivi ha permesso una comunicabilità elevatissima: le informazioni desiderate possono essere recuperate sempre ed ovunque in pochi minuti, grazie alla loro condivisione in reti quali internet.

Proprio per quanto riguarda lo scambio di informazioni col mondo esterno è necessario munire questi dispositivi di interfacce per il collegamento con l'esterno. L'utilizzo di queste interfacce influisce in modo negativo sull'autonomia del dispositivo consumando molta energia, ciò implica che bisogna porre dei limiti sul numero e sull'utilizzo delle interfacce per non costringere l'installazione di batterie troppo ingombranti e più pesanti.

Un esempio molto pratico e concreto si può fare su un dispositivo entrato oramai nel modo di vita di tutti noi, il cellulare. Lo sviluppo di questo dispositivo nato inizialmente solo per effettuare telefonate è vertiginoso, ora permette di inviare messaggi di testo, di effettuare fotografie, di personalizzare le suonerie, di scambiare e-mail, di navigare in internet, di giocare e tant'altro; tutto ciò però consuma energia: fotocamera nel caso delle fotografie o display grandi e spaziosi per i giochi; ciò non tanto a discapito dell'ingombro grazie alla miniaturizzazione della componentistica elettronica quanto all'eccessivo consumo energetico colpevole di un'autonomia delle batterie minore. Proprio per questi ultimi due elementi è invece richiesto l'opposto, cioè un ingombro sempre minore e un'autonomia maggiore; è importante trovare un compromesso, in quanto per aumentare l'autonomia sarebbe sufficiente installare batterie di dimensioni superiori

che però implica un aumento di peso del dispositivo non trascurabile.

Essendo dispositivi destinati ad un utilizzo mobile non sono accettabili i tradizionali collegamenti tramite cavo, le nuove tecnologie permettono comunicazioni senza cavo come infrarosso o onde radio; nello specifico Irda per la prima tipologia, Bluetooth e Wireless LAN per la seconda. A fronte di queste nuove tecnologie sono emerse nuove problematiche quali limitata copertura del servizio, tasso d'errore nella comunicazione maggiore che comporta la ritrasmissione delle informazioni perse, velocità di connessione minore e consumi di energia maggiori.

Proprio in questo quadro si colloca tale lavoro, il quale consiste in un nuovo modo di utilizzare una tecnologia esistente e consolidata quale la tecnologia wireless LAN, nella sua versione più diffusa è conosciuta, lo standard 802.11b; tale nuovo modo consente di minimizzare il consumo di energia per aumentare la durata delle batterie di cui ogni dispositivo portatile è fornito.

Generalmente ogni stazione wireless comunica con un server che può essere collegato ad internet, o disporre egli stesso delle informazioni richieste dal dispositivo portatile. Il nostro intento è la riorganizzazione (*reshaping*) del traffico, costituito dalle informazioni dirette verso il dispositivo portatile direttamente sul server, questo ci permette un risparmio energetico delle stazioni wireless senza modificare l'architettura attualmente esistente.

Sostenere una comunicazione costa molto in termini di energia consumata, per tale ragione è necessario evitare comunicazioni inutili come l'ascolto di traffico destinato ad altri o l'attesa continua di informazioni (*pacchetti*) che per qualche ragione non vi sono attualmente. A seguito degli elementi sopra elencati ci troviamo di fronte ad un problema di *Dinamic Power Management*, cioè di variare dinamicamente il consumo di energia della stazione wireless a seconda che debba o meno scambiare informazioni. Principalmente i metodi di power management cercano di mantenere la scheda wireless il più possibile in uno stato di basso consumo, perciò è importante il modo in cui le viene passato il traffico. Se un' applicazione invia pacchetti sporadicamente, tiene attiva per molto tempo la scheda wireless inutilmente; sarà allora necessario riorganizzare il traffico ad un livello sottostante, raggruppandolo ed inviandolo solo quando è veramente corposo, da rendere conveniente il risveglio della scheda.

A fronte delle considerazioni soprantanti, abbiamo deciso di spostare il problema dalle stazioni wireless al server principalmente per due motivi: innanzi tutto il server è collegato all'alimentazione di rete per cui non ha i problemi di autonomia delle batterie, e poi perchè ha la visione del traffico globale diretto verso tutte le stazioni wireless a lui collegate. Proprio tramite la visione globale del traffico è possibile un reshaping in modo che ogni stazione riceva solamente le informazioni a lei destinate o dirette a tutti (*broadcast*), cosa che allo stato attuale non è; infatti ogni stazione è sensibile anche alle informazioni dirette ad altri provocando un inutile consumo di energia. Il reshaping consiste nell'invio di blocchi di informazioni dirette verso una sola stazione, mettendola in stand-by quando non ve ne sono; attualmente esiste una sorta di modo di lavoro in stand-by analizzata nel dettaglio in seguito, ma al risveglio una stazione può ricevere comunque le informazioni altrui; il nostro intento è disaccoppiare il più possibile la visione delle informazioni altrui sino a farle scomparire.

A.2 CiuCiu

Prima di analizzare nel dettaglio come viene effettuato il reshaping è necessario vedere le caratteristiche principali del protocollo 802.11; proprio a tal fine il capitolo 2 è stato concepito per introdurre le problematiche relative allo scambio di informazioni in una rete senza fili e alla loro risoluzione. Inoltre sempre nel capitolo 2 viene mostrato lo stato attuale del protocollo da noi utilizzato, le implementazioni riguardanti il power management e la monitorizzazione del consumo di potenza.

Nel capitolo 3 verrà invece introdotto nel dettaglio il principio di funzionamento e l'idea che sta alla base dello scheduler da noi implementato per la riduzione del consumo energetico; inoltre verrà spiegata la relativa implementazione analizzando le singole sezioni del codice sorgente.

Nel capitolo 4 è riportato il modo in cui sono stati condotti i test, la realizzazione di una semplice architettura sperimentale per la monitorizzazione dei dati relativi al consumo di potenza e l'analisi dei dati da essa prodotti.

Infine nel capitolo 5 vi sono alcune considerazioni sul lavoro realizzato, i benefici introdotti e le ulteriori migliorie introducibili sempre in questa direzione.