Bollettino Unione Matematica Italiana

Sezione A – La Matematica nella Società e nella Cultura

Alessandro Avenali

Nuovi algoritmi per il problema dell'assegnazione delle frequenze di banda minima e per il problema dell'insieme stabile di peso massimo

Bollettino dell'Unione Matematica Italiana, Serie 8, Vol. **6-A**—*La Matematica nella Società e nella Cultura* (2003), n.2, p. 215–218.

Unione Matematica Italiana

<http://www.bdim.eu/item?id=BUMI_2003_8_6A_2_215_0>

Bollettino dell'Unione Matematica Italiana, Unione Matematica Italiana, 2003.

Bollettino U. M. I.
La Matematica nella Società e nella Cultura
Serie VIII, Vol. VI-A, Agosto 2003, 215-218

Nuovi algoritmi per il problema dell'assegnazione delle frequenze di banda minima e per il problema dell'insieme stabile di peso massimo.

ALESSANDRO AVENALI

Questa tesi è composta da due parti indipendenti.

1. – Problema dell'Assegnazione delle Frequenze di Banda Minima.

Nella prima parte della tesi si affronta il Problema dell'Assegnazione delle Frequenze di Banda Minima (Minimum Span Frequency Assignment Problem - MSFAP), che consiste nell'assegnare un adeguato numero di frequenze radio alle stazioni di base di una rete radio, in modo tale da minimizzare la banda di frequenze utilizzata e contemporaneamente garantendo che l'interferenza complessiva sia sotto una soglia prestabilita. MSFAP è un problema NP-hard in quanto il Problema della Colorazione del Grafo si riduce polinomialmente ad esso. MSFAP viene modellato con un grafo orientato, completo e pesato (nel seguito grafo-MSFAP), dove i nodi sono le stazioni di base, il peso d_v di ciascun nodo v è il numero di frequenze (interi non negativi) da assegnare alla corrispondente stazione di base, il peso (intero non negativo) di ogni arco (u, v) rappresenta la distanza che tutte le frequenze assegnate alle stazioni di base u e v devono avere affinchè siano rispettare le specifiche sull'interferenza; inoltre nel grafo c'è un nodo speciale s con peso unitario e con tutti gli archi ad esso incidenti di peso nullo. Si definisce *d-walk* un *walk* (sequenza di nodi) che inizia e finisce in s e che visita ogni nodo v esattamente d_v volte.

In [2] Janssen e Kilakos mostrano come un *lower bound* per MSFAP possa essere ottenuto calcolando il costo di un d-walk del grafo-MSFAP di costo minimo. Il problema di determinare tale costo viene formulato attraverso un problema lineare intero, dove variabili intere sono associate con gli archi del grafo-MSFAP.

Prendendo ispirazione da questo lavoro di Janssen e Kilakos, si definisce innanzitutto lo *span* di un walk w del grafo-MSFAP come il costo di un *subwalk* (sottosequenza di nodi) di w di costo minimo. Successivamente, rappresentando le assegnazioni di frequenze come walk del grafo-MSFAP, si dimostra che MSFAP è equivalente al problema di trovare un d-walk del grafo-MSFAP di *span* minimo.

Tale problema viene modellato con un opportuno problema lineare intero, associando variabili intere con particolari walk del grafo-MSFAP (più altre variabili intere necessarie alla gestione delle concatenazioni tra tali walk).

Tutti i risultati descritti possono essere trovati in [1]. Poichè il problema lineare intero proposto è caratterizzato da un numero esponenziale di variabili e vincoli, si definisce un metodo esatto basato su tecniche di *column and row generation* per la determinazione di una sua soluzione ottima (durante la sua esecuzione vengono generati sia lower bound che *upper bound* per MSFAP).

Inoltre tale l'algoritmo è basato su una strategia di risoluzione (rivelatasi molto utile per le grandi istanze) che consiste nel risolvere una sequenza di sottoproblemi del problema di partenza via via sempre più grandi e ciascuno strettamente contenuto nel successivo, e nell'estendere, attraverso una veloce euristica, ogni soluzione ammissibile trovata per il sottoproblema corrente a una soluzione ammissibile dell'intero problema.

L'algoritmo proposto risulta in grado di trovare soluzioni ottime su molti famosi benchmark di MSFAP (per esempio, le istanze di Philadelphia).

BIBLIOGRAFIA

[1] A. AVENALI, C. MANNINO e A. SASSANO, *Minimizing the Span of d*-walks to Compute Optimum Frequency Assignments, Math. Program., Ser. A, **91** (2002), 357-374.
[2] J. JANSSEN e K. KILAKOS, *Polyhedral Analysis of Channel Assignment Problems: (I) Tours*, Technical Report CDAM-96-17, London School of Economics (1996).

2. – Tabu Branch and Bound e il Problema dell'Insieme Stabile di Peso Massimo.

Nella seconda parte della tesi viene prima introdotto un problema combinatorio molto generale (tale problema è una generalizzazione di quello introdotto da Chvátal in [1]), e poi viene proposto un nuovo schema enumerativo per la risoluzione di tale problema.

Si definisce *clausola* un qualsiasi elemento dell'insieme $\{0, 1, *\}^n$ ($(*, \ldots, *)$ viene brevemente indicata con $\emptyset$), e *restrizione binaria* di una clausola y ogni clausola x tale che $x_i = y_i$ per ogni $i : y_i \neq *$ e $x_i \neq *$ per ogni $i : y_i = *$. Si indica con $T(y)$ l'insieme di tutte le restrizioni binarie di una clausola y; se $\Gamma = \{z^1, _p.., z^p\}$ è un insieme di clausole (eventualmente multiple), si definisce $T(\Gamma) = \bigcup_{i=1} T(z^i)$. Una clausola y tale che $T(\Gamma \cup \{y\}) \supset T(\Gamma)$ ($T(\Gamma \cup \{y\}) = T(\Gamma)$) è *nonredundant* (*redundant*) rispetto a Γ. Una funzione $f : \{0, 1, *\}^n \rightarrow \Re \cup \{-\infty, +\infty\}$ tale che $f(y) \geq f(x)$ per ogni clausola y e ogni $x \in T(y)$ viene definita *b-monotona*. Sia $f : \{0, 1, *\}^n \rightarrow \Re \cup \{-\infty, +\infty\}$ una funzione b-monotona. Il problema combinatorio oggetto di studio è definito come segue: $(f, \emptyset) = \{\max_{y \in T(\emptyset) = \{0, 1\}^n} f(y)\}$; tale problema (brevemente indicato con $(f, \emptyset)$) è NP-hard in quanto il Problema Lineare Binario si riduce polinomialmente ad esso.

Una classe molto nota di metodi per risolvere $(f, \emptyset)$ è quella fondata sul principio di *resolution* ([5]); tali algoritmi fanno uso di un'informazione elementare rappresentata dal concetto di *nogood*: dato un *incumbent* x^*, una clausola z è un *nogood* (rispetto a $[f, f(x^*)]$) se $f(x) \leqslant f(x^*)$ per ogni $x \in T(z)$ (un nogood rappresenta quindi un insieme di soluzioni ammissibili *tabù*, cioè non migliori dell'incumbent; in particolare se la clausola $\emptyset$ è un nogood allora x^* è una soluzione ottima). Un metodo è di tipo resolution se certifica l'ottimalità di x^* generando un insieme $\Gamma = \{z^1, \ldots, z^p\}$ di nogood tale che la clausola $\emptyset$ risulti redundant rispetto a Γ.

I nogood vengono tipicamente utilizzati per effettuare *fixing/pruning* su sottoproblemi del tipo $(f, y') = \{\max_{y \in T(y')} f(y)\}$, con y' una qualsiasi clausola; in particolare, dato un nogood z e un sottoproblema (f, y) si ha che: (i) se $y_i = z_i$ per ogni $i : z_i \neq *$ allora (f, y) deve essere scartato (*pruning*); (ii) se invece per ogni $i : z_i \neq *$, tranne uno e un solo indice j, $y_i = z_i$, allora la variabile y_j va fissata uguale a $1 - z_j$, in modo tale che dalla clausola y risultante non possa mai essere ottenuta, ponendo a 0/1 le eventuali variabili di y ad *, alcuna restrizione del nogood z (*fixing*).

Una famiglia di algoritmi di tipo resolution è quella basata su schemi di tipo *branch-and-bound*; esempi popolari sono il Branch and Bound (BB), il Backjumping Branch and Bound (BBB) e il Learning Branch and Bound (LBB).

Nell'albero di ricerca ogni nodo è un sottoproblema (f, y) e la clausola associata con ciascuna foglia dell'albero di ricerca risulta un nogood. I metodi basati su schemi di tipo branch-and-bound tipicamente si distinguono per come vengono trattati e poi utilizzati i nogood trovati. Per esempio nel BB non si fa alcun uso dei vari nogood; nel BBB invece si cerca di estrarre da ciascun nogood un altro nogood (il più corto possibile) che viene poi utilizzato per fare pruning e fixing sui sottoproblemi pendenti; nel LBB si opera come nel BBB ma in più i nogood vengono memorizzati nella speranza di poterli vantaggiosamente impiegare anche per sottoproblemi non ancora generati. Una caratteristica cruciale dell'approccio branch-and-bound è che ciascun nogood viene generato in maniera computazionalmente indipendente dai precedenti, cioè non è necessario tener traccia dei nogood già trovati per poter calcolare i successivi. Il principale incoveniente di tali algoritmi è che ciascun nogood deriva da un sottoproblema che è fortemente dipendente dall'albero di ricerca, cioè dalle *branching rule* (l'ordine con cui ogni nogood viene determinato dipende invece dalle *visiting rule*); pertanto i nogood trovati possono risultare viziati da inappropriate branching rule e di conseguenza pregiudicare l'efficacia del metodo (è un problema rilevante nel BB, un pò meno nel BBB e nel LBB).

Per superare tale ostacolo Chvátal ha proposto in [3] Resolution Search (RS), un algoritmo di tipo resolution in grado di generare nogood in maniera indipendente dall'albero di ricerca. Ciononostante, in RS ciascun nogood viene estratto da un sottoproblema (f, y) che è parzialmente prefissato da un meccanismo (automatico) necessario all'algoritmo per convergere; pertanto, anche in RS non si è completamente liberi di scegliere il sottoproblema da analizzare. Inoltre ogni nogood è determinato in modo da essere nonredundant rispetto a un opportuno insieme Γ di nogood, in tempo lineare nelle dimensioni di Γ (cioè $|\Gamma| \times L(\Gamma)$, dove

$1 \leqslant L(\Gamma) \leqslant n$ è la lunghezza del più lungo nogood in Γ); tale insieme Γ deve però contenere al più n dei nogood precedentemente calcolati.

Prendendo spunto da RS, viene proposto un nuovo algoritmo di tipo resolution, denominato Learning, tale che ciascun nogood (i) è generato a partire da un sottoproblema (f, y) arbitrariamente scelto (cioè indipendente sia dall'albero di ricerca che da altri meccanismi), e che inoltre (ii) risulta nonredundant rispetto a un insieme Γ con un numero qualsivoglia di nogood (sempre in tempo lineare nelle dimensioni di Γ).

La procedura Learning presenta però un incoveniente, in quanto necessita di inserire in Γ tutti i nogood precedentemente trovati; pertanto se $|\Gamma|$ cresce eccessivamente, trovare il prossimo nogood risulta un'operazione computazionalmente onerosa. Si propone allora un ulteriore algoritmo di tipo resolution, chiamato Tabu Branch and Bound (TBB), che consiste nell'utilizzare Learning finchè $|\Gamma| \leqslant m$, con m polinomiale, e poi nel completare l'enumerazione con uno schema di tipo branch-and-bound, in cui gli m nogood trovati vengono utilizzati per il pruning/fixing dei sottoproblemi generati e anche per definire una nuova regola di branching.

TBB è stato applicato per risolvere il Problema dell'Insieme Stabile di Peso Massimo (Maximum Weighted Stable Set Problem - MWSSP), che consiste nel trovare un insieme stabile (cioè di nodi mutuamente non adiacenti) di peso massimo in un grafo non orientato e pesato sui nodi.

Per calcolare gli upper bound per MWSSP si definisce un nuovo schema polinominiale, che consiste in una generalizzazione al caso pesato di una procedura (proposta da Mannino e Sassano in [4]) per il caso non pesato.

TBB risolve all'ottimalità molti DIMACS benchmark; inoltre risulta essere una buona euristica, in quanto in grado di trovare rapidamente delle buone soluzioni ammissibili (spesso ottime).

BIBLIOGRAFIA

[1] V. Chvátal., *Resolution Search*, Discrete Applied Mathematics, **73** (1997), 81-99.
[2] C. Mannino e A. Sassano, *An Exact Algorithm for the Maximum Stable Set Problem*, Computational Optimization and Application, **3** (1994), 243-258.
[3] J.A. Robinson, *A Machine-Oriented Logic Based on the Resolution Principle*, Journal of the Association for Computing Machinery, **12** (1965), 23-41.

Dipartimento di Informatica e Sistemistica «Antonio Ruberti»,
Università degli Studi di Roma «La Sapienza»
e-mail: avenali@dis.uniroma1.it
Dottorato in Ricerca Operativa (sede amministrativa:
Dipartimento di Statistica, Probabilità e Statistiche Applicate,
Università degli Studi di Roma «La Sapienza») - Ciclo XIV
Direttore di ricerca: Prof. Antonio Sassano,
Università degli Studi di Roma «La Sapienza»