

Università degli studi di Genova



Corso di Laurea in  
Economia marittima e dei trasporti



Il problema del cammino  
minimo in reti multiobiettivo

Relatrice: Anna Seimonachen

Candidato: Silvio Villa

*Ödicato a:*

*Coloro che in me  
Hanno sempre creduto  
E a coloro che  
Guardando oltre l'apparenza di  
Un abito hanno generosamente  
Elargito conforto, calore, amicizia.  
Va un grazie dovuto  
A quelle persone che mi hanno  
Restituito forza e audacia per  
Arrivare alla fine della corsa.*

# Introduzione

Questo lavoro tratta sinteticamente alcuni aspetti del problema del cammino minimo in reti urbane.

La ristrettezza degli spazi concessi ha fatto sì che il lavoro sia limitato ad un aspetto tipico del problema, a mio giudizio importante, ma non unico.

Poiché, secondo le più comuni interpretazioni della riforma dei corsi di laurea, appare che questo “breve elaborato scritto” debba rappresentare una forma di relazione finale con cui lo studente “chiude” il proprio ciclo di studi, ho ritenuto adeguato seguire, durante l’elaborazione di questa tesi, le linee guida apprese dal mio corso di studi.

Ho scelto di trattare una materia matematica per via del personale interesse verso la materia, trovando un ottimo mezzo per unire l’utile al dilettevole.

La ricerca operativa mi ha permesso di unire la mia passione verso la matematica al corso di studi svolto, applicando la stessa al settore dei trasporti.

Ho scelto altresì di fornire un’adeguata introduzione storica all’argomento, ritenendo particolarmente importante questa dimensione della matematica, spesso erroneamente trascurata.

Il primo capitolo introduce la materia, la teoria dei grafi ed il problema del cammino minimo, passando, dall’analisi del problema banale, attraverso l’introduzione di ipotesi e vincoli, allo studio di un problema che rappresenti, in modo il più possibile verosimile, la situazione reale.

Nel secondo capitolo, sono analizzati nel dettaglio tre algoritmi proposti da due ricercatori del dipartimento di ingegneria civile dell’università del Maryland; nel momento in cui questa tesi è scritta, il lavoro dei ricercatori americani è in fase di pubblicazione sul “European Journal of operational research”. Gli algoritmi in questione possiedono un certo grado di sofisticatezza matematica, tale da richiedere un’analisi approfondita; per questo motivo, e perché lo stesso lavoro rappresenta l’ultimo ritrovato in materia, ho preferito incentrarmi su que-

sta “promessa di pubblicazione”. L’innovazione proposta dai due scienziati è stata l’introduzione dell’analisi multicriterio in reti stocastiche e variabili rispetto al tempo.

Nel terzo capitolo sono passato ad analizzare il lavoro svolto da Paola Modesti e Anna Sciomachen nel 1998, che tratta un problema simile da un punto di vista leggermente diverso.

La mia tesi si risolve quindi con l’analisi delle criticità dei due lavori, con un confronto tra gli stessi e con le considerazioni personali sul problema dell’analisi multicriterio. Questa rappresenta infatti il fulcro del mio lavoro, in cui sono gettati i presupposti per possibili nuovi sviluppi dell’argomento, considerata l’opera di sintesi svolta attraverso la discussione dei lavori analizzati.

Con la speranza che questa conclusione possa tradursi nell’inizio di una nuova affascinante esperienza in ambito matematico.

*Silvio Villa, Genova, febbraio 2005.*

OPINION

UNO

Multicriteria Stochastic  
Time-Varying networks

# Capitolo I

## Multicriteria Stochastic Time-Varying networks

*Sommario: 1.1 L'origine della teoria dei grafi - 1.2 Introduzione alla teoria dei grafi - 1.3 Il problema del cammino minimo - 1.4 Algoritmi euristici - 1.5 Il problema del cammino minimo nelle reti urbane - 1.6 Introduzione al problema MSTV - 1.7 Relazioni tra i cammini Pareto-ottimali a priori e adattivi*

### 1.1 L'origine della teoria dei grafi

**b**enché già precedentemente si fosse ragionato in termini di grafi, la nascita della teoria dei grafi si può far risalire alla prima metà del secolo XVIII.

Leonhard Euler (1707 - 1783) introdusse alcuni simboli comunemente utilizzati quali  $e, \pi, i$ , elegantemente rappresentati insieme nella formula più affascinante della storia della matematica:  $e^{i\pi} + 1 = 0$ .

Oltre all'introduzione dei logaritmi di numeri negativi, Euler si rese protagonista della formalizzazione di una teoria di cui già Leibniz (Gottfried Wilhelm, 1646 - 1716) aveva parlato, tuttavia senza un preciso formalismo matematico: la teoria dei grafi.

Per oltre un secolo, questa teoria non fu ulteriormente sviluppata e si fermò al punto in cui era rimasta con gli studi di Euler; lo studio di questa teoria fu ripreso a partire dalla seconda metà del secolo XIX in Inghilterra, dove diversi problemi vennero formulati attraverso la teoria dei grafi.

Nel 1733, il gesuita Girolamo Saccheri (1677 - 1733) pubblica "*Euclides ab omni naevo vindicatus*", in cui, cercando di dimostrare il 5° postulato di Euclide per assurdo, getta le basi per le geometrie non euclidee.

Successivamente, Lambert (Johann Heinrich, 1728 - 1777) e Legendre (Adrien-Marie, 1752 - 1833) riprendono il lavoro di Saccheri, ma infruttuosamente; Gauss (Karl Friedrich, 1777 - 1855) nel 1824 afferma che una geometria basata sui primi 4 postulati di Euclide e sulla negazione del quinto è possibile e non contraddittoria, ma non pubblica i risultati dei suoi studi per paura delle reazioni del mondo matematico nei suoi confronti. Bisognerà aspettare il russo Lobachevskij (Nicolaj Ivanovič, 1793 - 1856) nel 1829 perché sia formalizzata la prima geometria non euclidea basata sulla negazione del quinto postulato: la geometria iperbolica. Nel XIX secolo, Riemann (Georg Friedrich Bernhard, 1826 - 1866) formula la geometria ellittica, postulando che due rette hanno sempre in comune almeno un punto, negando quindi l'esistenza di rette parallele.

In seguito, altri studi verranno effettuati sulle geometrie non euclidee che negano il quinto postulato, tra cui vale la pena ricordare il modello di Beltrami (Eugenio, 1835 - 1900), con cui lo studioso giunge alla definizione delle geodetiche, ed il modello di Klein (Felix, 1849 - 1925).

L'ironia della sorte volle che, mentre in tutti questi anni, gli sforzi di grandi matematici furono concentrati a ricercare una geometria che negasse il quinto postulato, la teoria dei grafi, già così come formulata da Euler rappresentava una geometria non euclidea; questa però partiva dalla negazione del terzo postulato, portando risultati per nulla trascurabili. In questa geometria, ad esempio, non è necessariamente vero che la somma di due lati di un triangolo sia maggiore del terzo lato.

Solo in epoca relativamente recente ci si è accorti del fatto che la teoria dei grafi rappresenta una geometria non euclidea, e più precisamente la prima ad essere stata formalizzata.

Tutto ebbe inizio nel 1735, nella cittadina di Königsberg (oggi Kaliningrad), nella Prussia orientale. La città, che diede i natali ad Immanuel Kant, è attraversata dal fiume Pregolya e un suo quartiere sorge su un'isola chiamata "der Kneiphof", oltre la quale, il fiume si divide in due diramazioni. A quei tempi, l'isola era collegata con ciascuna delle due sponde da due ponti, mentre la sponda situata dopo la suddivisione del fiume era collegata con un ponte per ogni sponda e con un altro all'isola.

Gli abitanti di Königsberg si chiedevano se esistesse un cammino che attraversasse tutti i ponti una ed una sola volta.

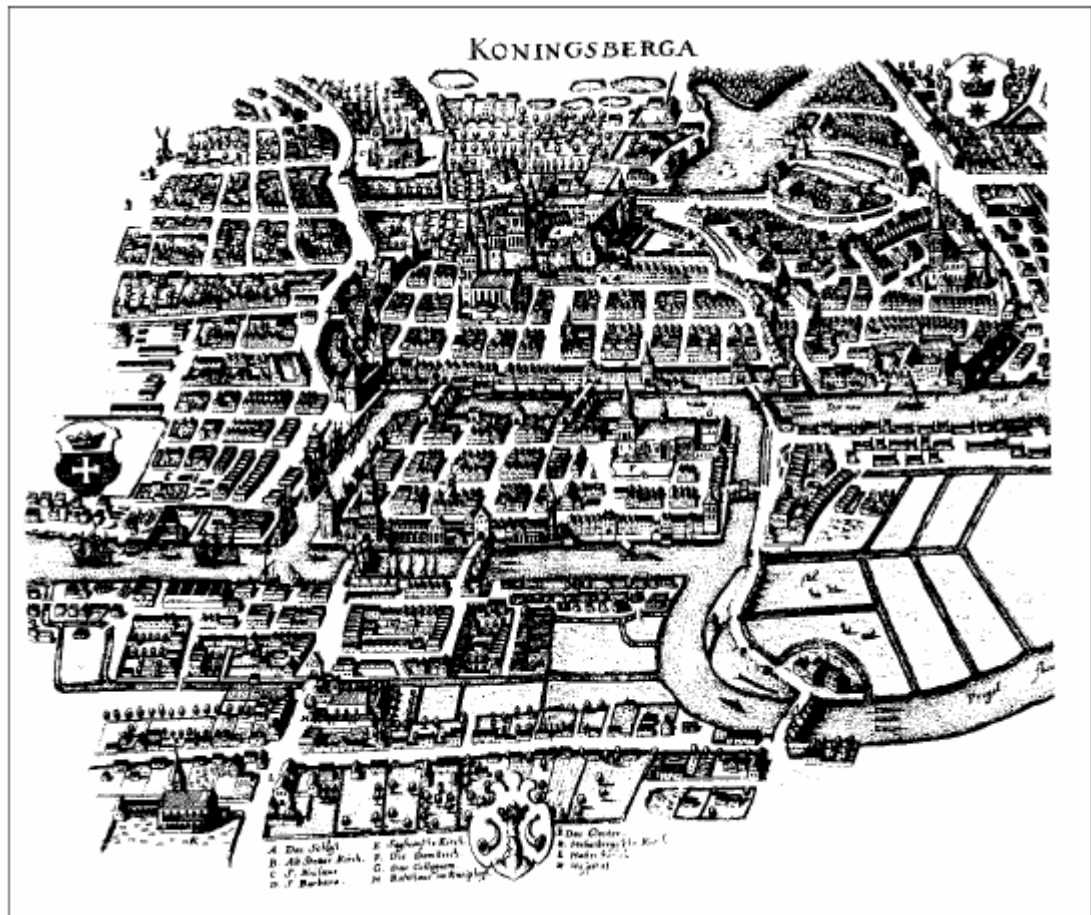


Fig. 1: Königsberg

Il quesito può essere affrontato direttamente per tentativi ma, vista la quantità elevata di possibili soluzioni, i cittadini non avrebbero comunque avuto la certezza di averli elencati tutti; quindi, se anche tutti i cammini elencati non risolveressero il problema, restava il dubbio di averli elencati tutti.

Generalizzando il problema, e ricorrendo alla modellizzazione possibile grazie alla teoria dei grafi, Euler riuscì ad uscire da quella situazione. Si pose infatti il problema di fornire una condizione generale di risolubilità per quel genere di problemi; fu quindi in grado di fornire la soluzione: il problema non ammette soluzioni.

Il risultato fu presentato all'Accademia di San Pietroburgo nell'agosto del 1736 e pubblicato nell'opera "*Solutio Problematis ad geometriam situs pertinentis*".



## 1.2 Introduzione alla teoria dei grafi

Un grafo è un insieme ordinato e finito di (punti) nodi legati tra loro da un insieme finito di relazioni (archi).

In questo tipo di geometria non valgono molte delle regole proprie della geometria euclidea; il concetto che risente maggiormente della negazione del terzo postulato di Euclide è quello di distanza. Qui non si può, infatti, parlare di "distanza euclidea".

Il concetto di distanza nella teoria dei grafi ha senso solo in termini numerici e non è di per se raffigurabile attraverso la rappresentazione grafica.

In termini formali si definisce grafo:

$\mathcal{G}(\mathcal{V}, \mathcal{E})$ , dove

$\mathcal{V}$  = insieme dei nodi  $\{v_1, v_2, \dots, v_n\}$

$\mathcal{E}$  = insieme degli archi  $\{e_1, e_2, \dots, e_m\}$

Un grafo si dice **orientato** se i suoi archi possono essere percorsi solo in una determinata direzione, mentre si dice **non orientato** se i suoi archi non hanno un senso di percorrenza e possono pertanto essere percorsi indifferentemente in entrambe le direzioni.

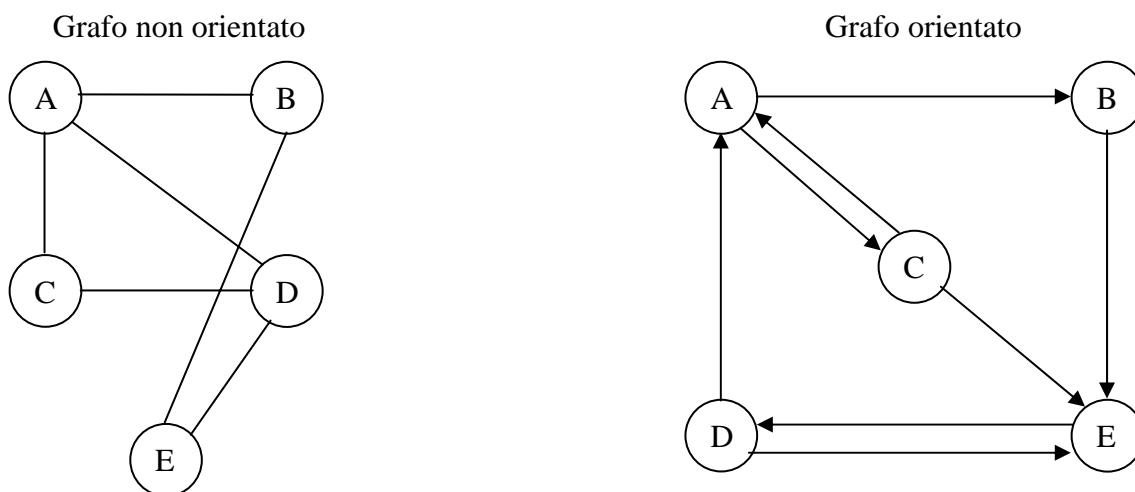


Fig. 2: Esempi di grafi

Due nodi si dicono **adiacenti** se sono connessi tra loro da un arco. Nell'esempio di Fig. 2, i nodi B ed E sono tra loro adiacenti, mentre i nodi B e D non lo sono.

Parallelamente, si dicono adiacenti due archi se hanno un nodo in comune; ad esempio, gli archi BE e AB sono adiacenti, mentre AB e DE non lo sono.

In un grafo orientato,  $\forall e_{ij} \in \mathcal{E}$ , il nodo  $i$  è detto **predecessore** del nodo  $j$ , ed il nodo  $j$  è detto **successore** del nodo  $i$ . Quindi, il nodo C è successore di A e predecessore di E.

Un grafo è **completo** quando esiste almeno un arco tra ogni coppia di nodi; viene definito **planare** se è possibile disegnarlo su un piano senza intersezioni tra gli archi.

Questa definizione è importante in ambito di reti urbane, poiché non possono rappresentarsi reti urbane non planari, in quanto ad ogni incrocio corrisponde un nodo, eliminando, di fatto, la possibilità che due archi possano intersecarsi.

Si parla inoltre di archi multipli, quando due nodi sono collegati tra loro da più di un arco, come in figura 3 a.

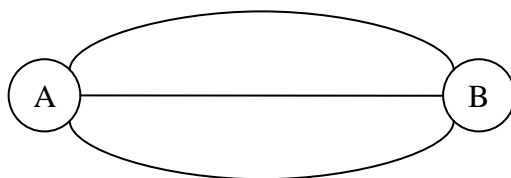


Fig. 3 a

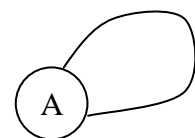


Fig. 3 b

I **loops** o **cappi**, sono archi per cui il nodo origine coincide con il nodo destinazione (fig. 3 b).

Un grafo che non contiene loops e archi multipli si definisce **semplice**.

Con il termine **cammino** si definisce una sequenza finita di archi nel quale il nodo finale di un arco è il nodo iniziale del successivo.

Se un cammino ha il nodo origine che coincide con il nodo destinazione, si ha un cammino chiuso, chiamato anche **circuito**.

Come già accennato, il concetto di distanza assume un significato particolare in questa geometria; un grafo si definisce **pesato** se ad ogni arco viene attribuito un valore definito "peso". Questo attributo può rappresentare il tempo necessa-

rio per percorrere l'arco, la distanza tra i due nodi, il costo per passare da un nodo all'altro, il flusso massimo consentito per unità di tempo, o qualsiasi altro significato.

La rappresentazione grafica di un grafo è visivamente molto efficace ed immediata ma, quando il numero di elementi, siano nodi o archi, è consistente, per rappresentare il grafo è preferibile servirsi di opportune matrici.

Le matrici più comunemente associate ai grafi sono generalmente tre:

La matrice di adiacenza  $A_{n \times n} = [a_{ij}]$  tale che:  $a_{ij} = \begin{cases} 1 & \text{se } (i, j) \in \mathcal{E} \\ 0 & \text{se } (i, j) \notin \mathcal{E} \end{cases}$

Se il grafo è pesato, viene definita la matrice dei pesi  $D_{n \times n} = [d_{ij}]$  tale per cui:

$d_{ij} = \begin{cases} w_{ij} & \text{se } (i, j) \in \mathcal{E} \\ +\infty & \text{se } (i, j) \notin \mathcal{E} \end{cases}$  con  $w_{ij}$  = peso associato all'arco  $(i, j)$

La matrice di incidenza  $B_{n \times m} = [b_{ia_{r,s}}]$  tale per cui:

$b_{ia_{r,s}} = \begin{cases} 1 & \text{se } i \text{ è nodo iniziale dell'arco } a_{r,s} \\ -1 & \text{se } i \text{ è nodo finale dell'arco } a_{r,s} \\ 0 & \text{se l'arco } a_{r,s} \text{ non incide sul nodo } i \end{cases}$

Il vantaggio di una rappresentazione attraverso le matrici consiste nella possibilità di applicare algoritmi iterativi o ricorsivi mediante l'uso di calcolatori elettronici.

Attraverso i grafi è possibile rappresentare molteplici tipi di situazioni: ad esempio una serie di attività legate da precise relazioni causa-effetto, anche se l'applicazione più comune e più intuitiva a cui questa geometria si presta è associata al problema del trasporto. Altri importanti problemi possono essere quello dell'allocazione di centri di servizio, oppure la progettazione di cicli di attività complesse quali quelle presenti nel campo dell'elettronica; recentemente, lo sviluppo delle reti neurali sta assumendo importanza sempre maggiore.

### 1.3 Il problema del cammino minimo

applicando la teoria dei grafi alle reti di trasporto, uno dei problemi più comuni è quello di trovare quel cammino che consenta di giungere da un certo nodo origine ad un nodo destinazione minimizzando il costo (la durata, la distanza).

In letteratura sono presenti diversi algoritmi per risolvere il problema, noto come **shortest path problem**, oltre alla possibilità di risolvere questo problema attraverso la programmazione lineare.

Il problema posto dalla programmazione lineare consiste nella sua limitata applicabilità a reti di piccole dimensioni, in quanto la complessità computazionale è elevatissima in termini di risorse necessarie per effettuare il calcolo.

Il modello di programmazione lineare necessita la matrice dei costi e la creazione di tante variabili binarie quanti gli archi esistenti che assumano valore 1 se l'arco rappresentato appartiene al cammino soluzione e 0 altrimenti; chiamando  $O$  il nodo origine e  $D$  il nodo destinazione, il modello consiste nella seguente espressione:

$$\begin{aligned} \min z &= \sum_{i,j} c_{ij} x_{ij} \\ \textcircled{+} \quad \sum_j x_{Oj} &= 1 \\ \textcircled{+} \quad \sum_i x_{iD} &= 1 \\ \textcircled{+} \quad \sum_i x_{ik} - \sum_j x_{kj} &= 0 \quad \forall k \neq O, D \\ \textcircled{+} \quad X(i, j) &\in \{0, 1\} \end{aligned}$$

In questa formulazione, il primo vincolo indica che, tra tutti gli archi uscenti dall'origine, solo uno può appartenere al cammino soluzione.

Al pari della condizione precedente, il secondo vincolo impone che uno solo degli archi entranti nel nodo destinazione possa essere attivo, quindi appartenere al cammino soluzione.

Il terzo vincolo impone che, se in un nodo  $k$  esiste un arco uscente, necessariamente deve esistere anche uno entrante.

Il quarto vincolo è la condizione che specifica che le variabili  $x_{ij}$  sono binarie.

È facile intuire come il problema assuma immediatamente dimensioni molto rilevanti dal punto di vista della complessità computazionale.

Esistono invece molteplici e differenti algoritmi in letteratura che risolvono il problema del cammino minimo, tra cui si possono distinguere diverse categorie.

Gli algoritmi arborescenti generano, attraverso una singola implementazione, l'albero dei percorsi minimi da un determinato nodo origine a tutti gli altri nodi. Applicando algoritmi di questo tipo a tutti i nodi assunti come origine, si ottiene una matrice, dove ogni applicazione dell'algoritmo costituisce la  $i$ -esima riga, che rappresenta il peso di ogni percorso minimo tra due nodi qualsiasi.

Generalmente gli algoritmi proposti fanno riferimento all'etichettatura dei nodi e si possono suddividere in due categorie a seconda che si tratti di algoritmi basati sull'attribuzione di etichette (label setting) o sulla correzione delle etichette (label correcting).

I primi ricercano, ad ogni iterazione, il percorso con il costo minimo dall'origine ad uno o più nodi della rete e, ad ogni passaggio, sono aggiornati con l'aggiunta di un nodo e del rispettivo arco che lo collega alla rete. L'algoritmo termina quando tutti i nodi sono stati connessi al nodo origine. La caratteristica di questo tipo di algoritmi è data dal fatto che ad ogni iterazione è fornito un risultato definitivo, cioè tutti i cammini tra l'origine ed i nodi attivi fino a quel momento sono ottimi. L'importanza di questa caratteristica si riflette in applicazioni pratiche, poiché non è necessario attendere il termine del processo per utilizzare già parte dei risultati.

I secondi invece effettuano l'aggiornamento ad ogni iterazione incrementando o modificando gli archi della soluzione, ma non garantiscono che il percorso ottenuto alla fine di ogni iterazione sia ottimo finché il processo non è terminato.

La distinzione tra queste categorie non è sempre netta, come nel caso del noto algoritmo di Dijkstra che è di tipo misto, cioè label setting-correcting.

Questo algoritmo appartiene alla categoria degli algoritmi “greedy” (cupidi), caratterizzati dalla scelta, ad ogni iterazione, della variabile a costo minimo, senza considerare se tale scelta porti o meno alla soluzione ottima. Per quanto riguarda l’algoritmo di Dijkstra, è dimostrato come per questo la soluzione generata sia sempre quella ottima.

Altri tipi di algoritmi sono in forma matriciale, e risolvono il problema del cammino minimo da ogni nodo della rete ad ogni altro nodo.

Il metodo maggiormente utilizzato è quello proposto da Floyd; questo, partendo dalla matrice dei costi, dopo ogni iterazione determina la lunghezza del cammino minimo tra ogni coppia di nodi.

## *1.4 Algoritmi euristici*

■ L’approccio esaustivo al problema, cioè l’utilizzo di un algoritmo che trovi l’insieme di tutte e sole le soluzioni ammissibili, necessita, come già detto, di un grande sforzo computazionale. Per questo motivo, si fa spesso ricorso a metodi definiti “euristici”.

Nei problemi di vehicle routing, si verificano spesso condizioni per cui risulta improponibile il ricorso a metodi esatti. L’approccio esaustivo a problemi di dimensioni realistiche è spesso assolutamente inefficiente; per problemi tali, cioè di dimensioni rilevanti, è necessario abbandonare il concetto di ottimalità assoluta e limitarsi a trovare soluzioni di buona qualità in un tempo convenientemente utile.

Il problema del cammino minimo è un problema semplice all’interno della teoria dei grafi; una complicazione di questo problema è data dal problema della distribuzione. Poiché non è interesse di questo lavoro il problema di distribuzione, diamo solo un accenno al problema ed ai metodi risolutivi utilizzati.

Il problema consiste nel raggiungere un certo numero di nodi prefissati da un punto di partenza utilizzando un dato numero di veicoli.

In questo tipo di problemi, per le ragioni viste sopra, difficilmente metodi esatti possono fornire soluzioni in tempi utili; questo ha portato alla nascita ed allo sviluppo di metodi euristici.

I metodi euristici si possono dividere in tre classi distinte: i metodi costruttivi, i metodi a due fasi e i metodi migliorativi.

I metodi costruttivi sono gli algoritmi più semplici, ma allo stesso tempo i più inefficienti. In tali algoritmi, la soluzione viene costruita in modo graduale, attraverso l'inserimento, ad uno ad uno, dei nodi nell'insieme soluzione, partendo da una situazione in cui nessun nodo è assegnato all'insieme soluzione. Il più noto algoritmo appartenente a questa categoria è il "Savings algorithm", noto anche come metodo Clarke & Wright, per il nome dei ricercatori che lo hanno proposto nel 1964.

I metodi in due fasi derivano dalla struttura del problema di distribuzione; infatti, gli algoritmi di questo tipo possono idealmente essere scissi in due fasi:

- ⊕ L'assegnazione dei nodi destinazione ai veicoli (clustering);
- ⊕ L'individuazione del ciclo ottimo per ogni insieme (routing);

Per trovare la soluzione ottima al problema, sarebbe necessario risolvere i due sottoproblemi contemporaneamente, mentre gli algoritmi in due fasi si propongono di risolvere separatamente i due problemi.

Esiste la possibilità di risolvere i due sottoproblemi nell'ordine inverso, cioè risolvere prima il problema di routing e, successivamente, quello di clustering; tuttavia, fino ad ora, questo secondo tipo di approccio non ha prodotto risultati soddisfacenti.

Un esempio di algoritmo appartenente a questa categoria è l'algoritmo "Sweep", come anche una versione modificata dell'algoritmo Clarke & Wright che considera separatamente le due fasi di clustering e routing.

I metodi migliorativi sono, a loro volta, un'ulteriore evoluzione degli algoritmi a due fasi; in questo tipo di algoritmi, viene superata la distinzione tra fase di clustering e fase di routing. Queste sono, infatti, applicate alternativamente, utilizzando, ciascuna, le informazioni provenienti dall'altra. Un algoritmo di que-

sta categoria, noto in letteratura, è quello proposto dai ricercatori canadesi Hertz, Gendreau e Laporte.

### *1.5 Il problema del cammino minimo nelle reti urbane*

**a**pplicando la teoria dei grafi alle reti urbane, le quali per loro natura sono congestionabili, si rendono opportuni necessari accorgimenti che comportano necessariamente implementazioni del problema e di conseguenza un aumento della difficoltà dello stesso.

Nelle reti di trasporto congestionabili, i tempi di percorrenza degli archi cambiano in funzione del momento della giornata, per via del numero di veicoli presenti nell'arco stesso.

Tuttavia, anche considerando la variabilità delle condizioni di traffico dovute ai cicli giornalieri, i tempi di percorrenza di un certo arco possono essere conosciuti a priori solo con incertezza, a causa di eventi imprevedibili quali incidenti, condizioni della strada comportamenti dei conducenti o, più in generale, fattori esterni.

In letteratura sono stati proposti molti approcci al problema del cammino minimo in reti in cui i pesi degli archi variano in funzione del tempo (Time Varying Networks), ma in modo deterministico (Cooke e Halsey, 1966; Dreyfus, 1969; Orda e Rom, 1990; Ziliaskopoulos e Mahmassani, 1993; Chabini, 1998). Questi lavori possono essere classificati come algoritmi basati sull'attribuzione o sulla correzione di etichette. Hall (1986), Miller-Hooks e Mahmassani (2000), e Pretolani (2000) studiarono problemi di cammino in reti in cui i pesi degli archi erano variabili aleatorie con funzioni di distribuzione variabili in funzione del tempo. Reti di questo tipo vengono definite Stochastic Time-Varying Networks (STV).

Essendo in presenza di reti in cui il peso degli archi è stocasticamente variabile nel tempo, non è possibile prevedere deterministicamente un cammino Pareto-



ottimale assoluto; inoltre, per questo motivo, sono possibili due differenti approcci al problema che, oltre a costituire due differenti impostazioni del problema, conducono a soluzioni non paragonabili tra loro, in quanto costituite da entità di ordine diverso.

In tutti questi lavori, vengono considerate infatti due classi di problemi: il problema del tempo medio atteso **a priori** (Least Expected Time, LET) e il problema del tempo medio atteso **adattivo**.

I problemi del primo tipo portano alla generazione di un cammino interamente determinato prima di intraprendere il viaggio, confrontando le possibili alternative e scegliendone una sulla base delle caratteristiche attese. Il risultato sarà quindi un singolo cammino, ovvero una sequenza di nodi.

I problemi del secondo tipo, al contrario, generano una serie di strategie di viaggio per cui il viaggiatore può, ad ogni nodo, scegliere la successiva direzione migliore in funzione del momento di arrivo in questo nodo. Tali strategie di percorso possono essere viste come ipercammini; un **ipercammino** è infatti l'insieme di tutti e soli i possibili cammini per arrivare dal nodo origine al nodo destinazione. Quindi una soluzione per questo tipo di problema non è data da una sequenza di nodi, bensì da una strategia di viaggio.

L'importanza assunta dagli ipercammini e dalle strategie adattive è dovuta al grande campo di applicazione di questo tipo di concetti: nelle reti urbane, i conducenti si trovano ad affrontare situazioni di questo tipo, in cui ad esempio un incidente può rendere completamente inefficiente una soluzione ottima a priori.

Fu e Rilett (1998) proposero un algoritmo euristico per il problema del cammino a priori.

Miller-Hooks (2001) diede un esempio per illustrare la differenza tra il problema a priori e il problema adattivo. Altri lavori che inseriscono il problema del cammino adattivo in reti stocastiche sono trattati da Polychronopoulos e Tsitsiklis (1996), Waller e Ziliaskopoulos (2003), Cheung (1998), Fu (2001) e Provan (2003).

Un'ulteriore implementazione al problema è data dalla considerazione di più di un criterio nella scelta del cammino minimo. Fino ad ora, sono stati presentati

lavori approfonditi solo per problemi con un singolo criterio. Per molte applicazioni può esserci più di un criterio per la scelta del cammino migliore: per esempio, pendolari, autisti di scuolabus, vettori di carichi pericolosi, preferiscono un cammino che simultaneamente minimizzi il costo del viaggio, il tempo di percorrenza, la distanza, la probabilità di incidenti, le esternalità negative sulla popolazione ed effetti simili.

Poiché è piuttosto improbabile che esista un singolo cammino minimo tra una data coppia di nodi origine-destinazione, ottimo secondo tutti i criteri considerati, la soluzione di un problema di cammino minimo nell'analisi multicriterio sarà una serie di cammini Pareto-ottimali (o non dominati).

Numerosi lavori propongono procedure di soluzione per problemi di cammino multicriterio, in cui tutti gli attributi degli archi sono assunti in modo deterministico e invarianti rispetto al tempo. Climaco e Martins (1982) proposero un algoritmo basato sull'idea di partire dai  $k$  percorsi più corti per risolvere il problema del cammino minimo con due criteri (bicriterio).

Martins (1984) sviluppò due algoritmi per generare tutti i cammini Pareto-ottimali. Un algoritmo è la generalizzazione dell'approccio label-setting di Hansen per questo problema. Corley e Moon (1985) svilupparono un algoritmo basato sulla correzione delle etichette per generare tutti i cammini Pareto-ottimali. Zografos e Davis (1989) impiegarono la programmazione di meta per progettare il viaggio ottimo multiobiettivo di materiali pericolosi in una rete statica. Nel contesto dell'assegnazione del traffico, Dial (1979) propose una tecnica per generare scelte di metodi di instradamento Pareto-ottimali.

Nonostante le tecniche o l'applicazione considerata, la generazione di tutti i cammini Pareto-ottimali può richiedere la generazione di tutti i possibili cammini, poiché tutti i cammini possono essere Pareto-ottimali. Quindi, ogni tecnica che generi tutte le soluzioni Pareto-ottimali nella peggiore delle ipotesi ha una complessità computazionale che cresce in modo esponenziale.

A causa di questa complessità, alcuni ricercatori hanno applicato funzioni di utilità per studiare problemi multidimensionali di cammino minimo in reti statiche. Modesti e Sciomachen (1998) usarono l'algoritmo di Dijkstra per determi-

nare i cammini che minimizzino funzioni di utilità lineari in reti deterministiche per risolvere un problema multiobiettivo di assegnazione del traffico.

Loui (1983) ed Eiger (1985) mostrarono come, quando la funzione di utilità è lineare o esponenziale, ogni algoritmo basato sulle etichette può essere usato per trovare il cammino ottimo in reti statiche senza violare il principio di Bellman. Carraway (1990) propose una generalizzazione dell'algoritmo di programmazione dinamica basato sul principio debole di ottimalità per l'utilizzo con funzioni di utilità non monotone. Il suo approccio generalizzato alla programmazione dinamica inserisce il problema del cammino minimo multicriterio nelle reti statiche e acicliche quando il principio di Bellman può essere violato.

Mirchandani e Wiecek (1993) ridussero il problema stocastico del cammino minimo al problema del cammino ottimo multiattributo con una funzione di utilità monotona non lineare.

Mirchandani e Soroush (1985) svilupparono un algoritmo efficiente per risolvere il problema del cammino ottimo in una rete stocastica con una funzione di utilità parabolica.

Alcuni lavori studiarono problemi di cammino multicriterio in reti stocastiche a tempi costanti (time-invariant). Turnquist (1987) suggerì l'utilizzo della simulazione insieme ad algoritmi basati sulle etichette per risolvere questi problemi.

Supposti gli attributi degli archi distribuiti normalmente, Wijeratne (1993) sviluppò l'algoritmo di cammino minimo stocastico e multiobiettivo (Stochastic Multiobjective Shortest Path) per trovare l'insieme di cammini in questo tipo di reti. Il problema è esteso al multicriterio, ma i criteri sono ridotti a due fattori deterministici; quindi il problema finale è ridotto ad un problema deterministico multiobiettivo.

Tutti i lavori precedentemente elencati assumono che gli attributi degli archi siano costanti rispetto al tempo (time-invariant). Esistono comunque diverse applicazioni per cui multipli attributi, come il tempo di percorrenza, il costo di percorrenza, le esternalità negative sulla popolazione ed il numero di incidenti, possono variare in funzione del tempo (time-varying). Nozick (1997) sviluppò un approccio integrato di pianificazione e instradamento per risolvere un problema multicriterio legato a spedizioni di carichi pericolosi con attributi variabi-

li in funzione del tempo ma deterministici. Le soluzioni risultanti formano un insieme di combinazioni di tempi di partenza-viaggio. Questo approccio non può garantire che siano generati tutti i cammini Pareto-ottimali. In reti STV, Miller-Hooks e Mahmassani (1998) fornirono algoritmi di tipo label-correcting per generare tutti i cammini Pareto-ottimali a priori nel rispetto di diverse definizioni di preferenza. Chang è in procinto di presentare un algoritmo euristico per risolvere problemi di cammino multicriterio in reti STV in cui si assume che tutti gli attributi di tutti gli archi siano variabili aleatorie continue.

## *1.6 Introduzione al problema MSTV*

**S**

embrebberebbe che non vi siano pubblicazioni che considerino la generazione di tutti i cammini adattivi Pareto-ottimali in questo contesto.

I ricercatori Sathaporn Opananon e Elise Miller-Hooks hanno proposto un lavoro, che al momento attuale è in fase di pubblicazione, in cui è presentato un algoritmo che genera tutte le strategie adattive Pareto-ottimali (riferite a ipercammini Pareto-ottimali) in reti STV con attributi multipli sugli archi, cioè in reti STV multicriterio (MSTV).

Specificamente, sono cercati i cammini che minimizzino il valore atteso secondo più criteri da tutte le origini ad una specificata destinazione per tutti i tempi di partenza in un intervallo di tempo definito. Questi ipercammini includono un insieme di strategie che forniscono al viaggiatore le direzioni Pareto-ottimali da prendere ad ogni nodo in funzione del momento di arrivo nelle locazioni intermedie. Sebbene un decisore potrebbe, a posteriori, scegliere una singola soluzione ottima tra tutti gli ipercammini Pareto-ottimali, la generazione di tutti questi ipercammini potrebbe richiedere enormi sforzi computazionali. Così, si presenta un algoritmo per generare efficientemente un singolo ipercammino ottimo rappresentando esplicitamente i criteri di preferenza del decisore attraverso una funzione di disutilità (costo generalizzato). Queste tecniche conducono

verso la reale complessità nella selezione del cammino considerando le variabili condizionanti il viaggio, cioè considerano la natura multiobiettivo di molte decisioni relative alla scelta di un cammino.

Diamo ora un esempio per illustrare la natura ed i vantaggi degli ipercammini soluzione nelle reti MSTV. Similmente ai problemi di cammino in reti STV con un singolo attributo associato ad ogni arco, quando esistono più attributi variabili in funzione del tempo, possono essere prese decisioni corrette solo scegliendo il cammino in modo adattativo. In figura 4, è illustrata una rete MSTV con due attributi per arco, cioè il tempo ed il costo di percorrenza. Assumiamo che entrambi siano variabili in funzione del tempo e siano conoscibili solo probabilisticamente. L'attesa al nodo non è permessa e si assume che gli attributi degli archi siano indipendenti nello spazio e nel tempo e siano indipendenti reciprocamente l'un l'altro.

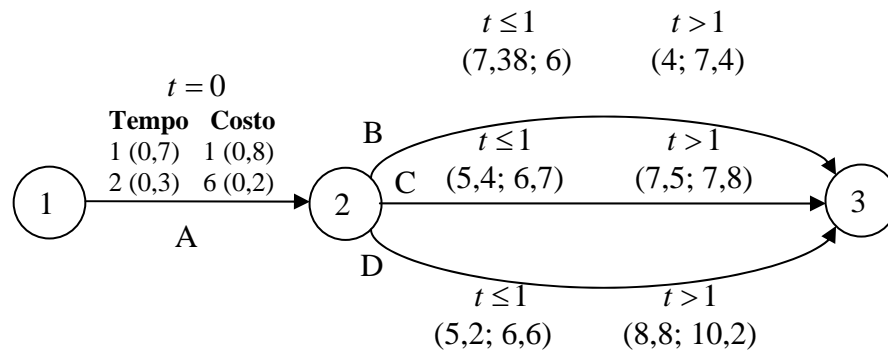


Fig. 4

Gli attributi sono espressi o come valori attesi (nei sottocammini B, C, D) o come variabile aleatoria discreta, in cui la probabilità di ogni possibile risultato è data tra parentesi (per l'arco A). Per esempio, ci sono due possibili tempi di percorrenza per l'arco A partendo dal nodo 1 con  $t=0$ : 1 con probabilità 0,7 e 2 con probabilità 0,3. Ci sono anche due possibili costi: 1 con probabilità 0,8 e 6 con probabilità 0,2.

Per semplicità, sono forniti direttamente i valori attesi per ogni attributo dei sottocammini B, C e D; per esempio, il tempo ed il costo di percorrenza attesi per il cammino B con  $t \leq 1$  sono 7,38 e 6. Si può osservare che esistono tre cammini

(A-B, A-C, A-D) tra il nodo origine ed il nodo destinazione. Supponiamo che il viaggiatore parta dal nodo 1 al tempo  $t=0$ : gli attributi attesi per questi tre archi possono essere trovati come segue:

P<sub>1</sub>) Cammino A-B

Tempo di percorrenza atteso:  $(1+7,38) \cdot 0,7 + (2+4) \cdot 0,3 = 7,67$

Costo di percorrenza atteso:

$(1+6) \cdot 0,7 \cdot 0,8 + (6+6) \cdot 0,7 \cdot 0,2 + (1+7,4) \cdot 0,3 \cdot 0,8 + (6+7,4) \cdot 0,3 \cdot 0,2 = 8,42$

Impiegando lo stesso procedimento si determinano anche i tempi ed i costi di percorrenza attesi degli altri possibili cammini, che risultano essere per A-C: (7,33; 9,03) e per A-D: (7,58; 9,68).

La dominanza ad un particolare istante di partenza può essere stabilita da criteri di confronto dei valori attesi, sia per il problema a priori che per quello adattivo. Così, per  $t=0$  come istante di partenza, esistono due cammini Pareto-ottimali a priori: il cammino A-B e A-C, poiché il cammino A-D è dominato dal cammino A-C.

Nella versione adattiva di questo problema di cammino minimo multicriterio, il viaggiatore può posticipare la sua scelta tra i sottocammini B, C e D fino al momento del suo arrivo al nodo 2. Il tempo atteso per una soluzione adattivo si calcola come:

H<sub>1</sub>) Cammino A-B se l'arrivo al nodo 2 si ha con  $t=1$  o cammino A-C se l'arrivo al nodo 2 si ha ad un istante  $t > 1$

Tempo di percorrenza atteso:  $(1+7,38) \cdot 0,7 + (2+7,5) \cdot 0,3 = 8,72$

Costo di percorrenza atteso:

$(1+6) \cdot 0,7 \cdot 0,8 + (6+6) \cdot 0,7 \cdot 0,2 + (1+7,8) \cdot 0,3 \cdot 0,8 + (6+7,8) \cdot 0,3 \cdot 0,2 = 8,54$

Impiegando lo stesso procedimento si determinano anche i tempi ed i costi di percorrenza attesi per ogni possibile ipercammino per questo problema. Questi sono riassunti nella seguente tabella.

Indice dell'ipercammino	Strategia risultante dal tempo di arrivo al nodo 2		(Tempo atteso, costo atteso)
	$t = 1$	$t = 2$	
H <sub>1</sub>	B	C	(8,72; 8,54)
H <sub>2</sub>	C	B	(6,28; 8,91)
H <sub>3</sub>	B	D	(9,11; 9,26)
H <sub>4</sub>	D	B	<b>(6,14; 8,84)</b>
H <sub>5</sub>	C	D	(7,72; 9,75)
H <sub>6</sub>	D	C	(7,19; 8,96)
H <sub>7</sub>	B	B	<b>(7,67; 8,42)</b>
H <sub>8</sub>	C	C	(7,33; 9,03)
H <sub>9</sub>	D	D	(7,58; 9,68)

Nota: gli ipercammini H<sub>7</sub>, H<sub>8</sub>, H<sub>9</sub> sono identici rispettivamente alle soluzioni a priori A-B, A-C e A-D.

Solo H<sub>4</sub> e H<sub>7</sub> sono non dominate. Entrambe le soluzioni comportano che il viaggiatore percorra l'arco A con  $t = 0$ . Poiché l'attesa non è permessa, entrambe le soluzioni indicano che la mossa successiva dal nodo 2 sia il sottocammino B se il tempo di arrivo è 2. Con tempo di arrivo 1, si può scegliere sia il sottocammino D, sia il sottocammino B. Occorre notare che, per questo esempio, Il cammino Pareto-ottimale A-C per il problema a priori è dominato da H<sub>4</sub>.

### *1.7 Relazioni tra i cammini Pareto ottimali a priori e adattivi.*

**g**li autori propongono ora alcune proposizioni che mostrano le relazioni esistenti tra i cammini ottimi a priori e quelli adattivi.

*Proposizione 1:* Una soluzione Pareto-ottimale al problema adattivo non può essere dominata da nessuna soluzione a priori.

*Discussione:* Ciò è conseguenza del fatto che ogni cammino a priori può servire un ipercammino.

Nel problema del cammino minimo adattivo a criterio singolo, per un dato istante di partenza da una data origine, il tempo atteso dell'ipercammino soluzione fornisce un margine inferiore per il tempo atteso del cammino minimo a

priori (la dimostrazione è di Miller-Hooks e Mahamassani, 2000). Mostriamo immediatamente che questo concetto non è necessariamente vero quando vengono considerati più criteri.

*Proposizione 2:* Una soluzione Pareto-ottimale al problema del cammino adattivo può non contenere neanche un criterio per cui il suo valore atteso sia minore o uguale a tutte le soluzioni a priori.

*Dimostrazione:* Assumiamo per assurdo che un ipercammino Pareto-ottimale debba contenere almeno un criterio per cui il suo valore atteso sia minore o al più uguale a tutte le soluzioni a priori. Per fornire un esempio contrario, possiamo modificare la rete presentata nell'esempio, cioè cambiare il tempo di percorrenza del sottocammino C per  $t > 1$  da 7,5 a 3,95. Con questa correzione, le tre soluzioni a priori hanno i seguenti valori attesi per i due criteri:  $P_1$  (7,67; 8,42),  $P_2$  (6,27; 9,03),  $P_3$  (7,58; 9,68).

Per lo stesso esempio si possono trovare quattro ipercammini Pareto-ottimali:  $H_1$  (7,65; 8,54),  $H_4$  (6,14; 8,84),  $H_6$  (6,13; 8,84),  $H_7$  (7,67; 8,42). L'ipercammino soluzione  $H_1$  è Pareto-ottimale e non contiene un criterio per cui il suo valore atteso sia minore o uguale a tutte le soluzioni a priori, contraddicendo l'assunto.

*Proposizione 3:* Il minor valore atteso di tutte le soluzioni Pareto-ottimali al problema del cammino adattivo per ogni criterio è minore o al più uguale a quello di ogni soluzione a priori.

*Dimostrazione:* Per ogni criterio, se esiste una soluzione a priori il cui valore atteso per quel criterio è il minore di tutti gli ipercammini soluzione Pareto-ottimali, questa soluzione a priori sarebbe Pareto-ottimale per il problema adattivo (cioè sarebbe un ipercammino Pareto-ottimale).

*Proposizione 4:* Una soluzione Pareto-ottimale al problema a priori può non contribuire a nessuna soluzione Pareto-ottimale al problema del cammino adattivo.

*Discussione:* La rete mostrata in figura 4 fornisce un esempio contrario. Sebbene la soluzione a priori A-C sia non dominata per il problema a priori, quando possono essere prese decisioni adattive, non è mai ottimo continuare dal nodo 2



lungo il sottocammino C, poiché tutte le soluzioni adattive che contengono il sottocammino C sono dominate.

*Proposizione 5:* Un cammino dominato nel problema a priori può contribuire alla soluzione del problema del cammino adattivo.

*Discussione:* Nuovamente, la rete di esaminata fornisce un contro esempio. Il cammino A-D è dominato nel problema a priori. Comunque, quando sono permesse soluzioni adattive, il cammino A-D è una strategia Pareto-ottimale per un dato istante di partenza dal nodo 2, quindi contribuendo alla soluzione Pareto-ottimale del problema del cammino adattivo.

OPRIMO

DVS

Algoritmi esatti

# Capitolo 2

## Algoritmi esatti

*Sommario: 2.1 Notazione nelle reti e definizioni del problema - 2.2 L'algoritmo APS - 2.3 Considerazioni sull'algoritmo APS - 2.4 L'algoritmo ABCDS I - 2.5 Considerazioni sull'algoritmo ABCDS I - 2.6 L'algoritmo ABCDS II - 2.7 Considerazioni sull'algoritmo ABCDS II - 2.8 Esperimenti su convenienti reti standard - 2.9 Conclusioni*

### *2.1 Notazione nelle reti e definizioni del problema*

**S**ia  $G = (\mathcal{V}, \mathcal{A}, \mathcal{S}, \mathcal{C}, \mathcal{P}, \mathcal{R})$  un grafo finito, dove  $\mathcal{V}$  è l'insieme dei nodi e  $\mathcal{A}$  è l'insieme degli archi direzionati che connettono i nodi.  $\Gamma^{-1}(i)$  rappresenta l'insieme dei nodi predecessori del nodo  $i$ , cioè tutti i nodi  $j$  tali per cui l'arco  $(i, j) \in \mathcal{A}$ .

L'intervallo di interesse, riferito al periodo di punta, è ridotto a discreto in piccoli intervalli di tempo rappresentati da  $S = \{t_0 + s\Delta t\}_{s=\{1,2,3,\dots,l\}}$ , dove  $\Delta t$  è la lunghezza di ogni intervallo di tempo.

Dopo l'ultimo periodo, si assume che gli attributi degli archi siano stazionari, prendendo gli stessi valori assunti nell'ultimo intervallo di tempo,  $t_0 + l\Delta t$ . Vengono associati diversi attributi ad ogni arco; assumiamo che questi attributi siano variabili aleatorie discrete con funzione densità di probabilità data dall'insieme  $(\mathcal{C}, \mathcal{P})$ , cioè l'insieme degli attributi degli archi e le corrispondenti probabilità di verificarsi, dove l'insieme  $\mathcal{R} = \{1, 2, \dots, r\}$  indica il corrispondente criterio.

Per ogni arco  $(i, j) \in \mathcal{A}$ ,  $k \in \mathcal{R}$ ,  $C^k = \{c_{ij}^{kz_k}(t)\}_{\text{con } z_k=1, \dots, D}$  indica l'insieme dei  $D$  possibili valori degli archi per il criterio  $k$  per attraversare l'arco con partenza all'istante  $t$ .

Per ogni criterio preso in considerazione, esistono  $D$  possibili valori che ogni arco può assumere.  $z_k$  rappresenta il numero d'ordine del valore assumibile dalla variabile aleatoria.

Per ogni possibile valore dell'arco, si assume che  $c_{ij}^{kz_k}(t)$  sia un valore reale non negativo con probabilità di verificarsi  $\rho_{ij}^{kz_k}(t) \in \mathcal{P}^k$ .

Nel contesto dell'assistenza per il cammino minimo, la durata del viaggio è un criterio che viene spesso considerato. Quindi, assumiamo che il tempo di viaggio sia il primo degli  $r$  criteri considerati, cioè in questo lavoro per  $k=1$  si intende il criterio della durata del viaggio.

I valori degli attributi degli archi e le corrispondenti probabilità sono definite all'in-gresso dell'arco e si considerano statiche per quel particolare viaggiatore fino all'uscita dell'arco. Questa condizione viene generalmente chiamata "proprietà dell'arco congelato" (Orda e Rom, 1990).

Per ogni istante di partenza successivo al periodo di punta, cioè per  $t > t_0 + I\Delta t$ ,  $c_{ij}^{kz_k}(t) = c_{ij}^{kz_k}(t_0 + I\Delta t)$  e allo stesso modo  $\rho_{ij}^{kz_k}(t) = \rho_{ij}^{kz_k}(t_0 + I\Delta t)$ ,  $\forall k, z_k, i, j$ ; questa ipotesi è dovuta al fatto che, al di fuori del periodo di punta, gli attributi possono essere considerati ragionevolmente meno variabili rispetto al tempo. Assumiamo che gli attributi degli archi siano indipendenti nello spazio e nel tempo e siano indipendenti tra loro.

Il lavoro preso in esame tratta due ordini di problemi differenti:

- ⊗ Cercare tutte le strategie di ipercammino Pareto-ottimali, nel rispetto dei valori attesi di ogni criterio, da ogni nodo a destinazione, per ogni istante di partenza  $t \in S$ .
- ⊗ Cercare un singolo ipercammino ottimo nel rispetto di una funzione lineare di disutilità da ogni nodo ad ogni specifica destinazione per ogni istante di partenza  $t \in S$ .

Per quanto riguarda il primo problema, sia  $H(t)$  l'insieme di tutti i possibili ipercammini che collegano una coppia origine-destinazione per l'istante di partenza  $t$  e sia  $\tilde{\theta}_a^k(t)$  la variabile aleatoria per il  $k$ -esimo criterio lungo l'ipercammino  $a \in H(t)$ .

$E[\tilde{\theta}_a(t)] = \{E[\tilde{\theta}_a^1(t)], E[\tilde{\theta}_a^2(t)], \dots, E[\tilde{\theta}_a^k(t)], \dots, E[\tilde{\theta}_a^r(t)]\}$ , dove  $E[\tilde{\theta}_a^k(t)]$  è il valore atteso della variabile aleatoria  $\tilde{\theta}_a^k(t)$ . Dati un nodo  $i \in \mathcal{V}$  e un istante di partenza  $t \in \mathcal{S}$ , l'ipercammino  $a$  è Pareto-ottimale se non esiste nessun altro ipercammino  $b \in H(t)$  tale per cui

$$E[\tilde{\theta}_b^k(t)] \leq E[\tilde{\theta}_a^k(t)] \quad \forall k \in \{1, 2, \dots, r\} \quad \text{ed esiste } h \in \{1, 2, \dots, r\} \quad \text{tale per cui} \\ E[\tilde{\theta}_b^h(t)] < E[\tilde{\theta}_a^h(t)].$$

Nel secondo tipo di problema, l'ipercammino  $f \in H(t)$  rappresenta l'ipercammino con la minore disutilità attesa per l'istante di partenza  $t$  se  $E[U_f(t)] = \min_{g \in H(t)} E[U_g(t)]$ , dove  $E[U_g(t)] = \sum_{k=1}^r w^k \cdot E[\tilde{\theta}_g^k(t)]$  e  $w^k$  è il peso assegnato al criterio  $k$ .

## 2.2 L'algoritmo APS

**n**ell'algoritmo del limite atteso più basso (ELB), prima della conclusione, ad ogni nodo è associata un'etichetta per ogni istante di partenza, ciascuna delle quali rappresenta un limite superiore al tempo medio di percorrenza dal nodo in questione alla destinazione per quel determinato istante di partenza.

Terminato il processo, ogni etichetta fornisce il minor tempo medio di percorrenza verso la destinazione desiderata. Diversamente dalle soluzioni del problema adattivo a criterio singolo, in cui esiste un singolo ipercammino soluzione per ogni nodo e per ogni istante di partenza, nel problema adattivo multicri-

terio possono esistere diversi ipercammini soluzione Pareto-ottimali per ogni nodo ed istante di partenza.

A maggior ragione, invece che calcolare un singolo valore atteso, dovranno essere mantenuti  $\mathcal{R}$  valori attesi, cioè uno per criterio. Similmente al problema a criterio singolo, dove un particolare ipercammino può essere ottimale solamente dato un particolare istante di partenza, nel problema multiobiettivo un ipercammino Pareto-ottimale per un certo istante di partenza può essere dominato per un altro istante di partenza. Il calcolo di questi ipercammini richiede la conoscenza delle informazioni relative al cammino solo per gli istanti di partenza in cui questi ipercammini non sono dominati. Quindi, è sufficiente mantenere solo gli ipercammini per gli istanti di partenza in cui sono non dominati.

Per ogni nodo  $i \in \mathcal{V}$ , ogni istante di partenza  $t \in \mathcal{S}$  ed ogni ipercammino Pareto-ottimale  $x$  è mantenuto un vettore etichetta  $\lambda_{ix}(t) = \{\lambda_{ix}^k(t)\}_{con\ k \in \mathcal{R}}$  in cui ogni elemento  $\lambda_{ix}^k(t)$  del vettore è il valore atteso rispetto al criterio  $k$  lungo l'ipercammino  $x$  dal nodo  $i$ , ad un certo istante di partenza  $t$ , al nodo destinazione  $N$ . Finché non ha termine l'algoritmo, sono mantenuti più vettori etichetta per ogni nodo e per ogni istante di partenza. Nelle iterazioni intermedie dell'algoritmo,  $X_i(t)$  contiene le etichette degli ipercammini soluzione correnti per il nodo  $i$  all'istante di partenza  $t$ .

$|X_i(t)|$  è uguale al numero di etichette vettore attualmente mantenute per il nodo  $i$  dato il tempo  $t$ .

Si assume che  $\forall t \in \mathcal{S}, i \in \mathcal{V}, k \in \mathcal{R}, 0 < \varepsilon < \Delta t, \lambda_{ix}^k(t + \varepsilon) = \lambda_{ix}^k(t)$ .

Inoltre, per ogni istante di partenza successivo al periodo di punta, per la proprietà dell'arco congelato,  $t > t_0 + I\Delta t, \lambda_{ix}^k(t) = \lambda_{ix}^k(t + I\Delta t)$ .

Di seguito è illustrata la struttura dei vettori etichetta per ogni istante di partenza e si dimostra che può essere mantenuto più di un vettore etichetta, ciascuno associato a differenti ipercammini, per ogni istante di partenza. Rendiamo discreto il periodo di interesse assumendo  $\Delta t = 1$  e assumendo che ci siano due ipercammini Pareto-ottimali per  $t = 0, 1, 3$  ed uno solo per  $t = 2$ .

Per cui,  $X_i(0) = X_i(1) = X_i(3) = \{1, 2\}$  e  $X_i(2) = \{1\}$ .

Tempo	Posizione 1	Posizione 2
0	$\lambda_{i1}(0) = \{\lambda_{i1}^1(0), \lambda_{i1}^2(0), \dots, \lambda_{i1}^r(0)\}$	$\lambda_{i2}(0) = \{\lambda_{i2}^1(0), \lambda_{i2}^2(0), \dots, \lambda_{i2}^r(0)\}$
1	$\lambda_{i1}(1) = \{\lambda_{i1}^1(1), \lambda_{i1}^2(1), \dots, \lambda_{i1}^r(1)\}$	$\lambda_{i2}(1) = \{\lambda_{i2}^1(1), \lambda_{i2}^2(1), \dots, \lambda_{i2}^r(1)\}$
2	$\lambda_{i1}(2) = \{\lambda_{i1}^1(2), \lambda_{i1}^2(2), \dots, \lambda_{i1}^r(2)\}$	-
3	$\lambda_{i1}(3) = \{\lambda_{i1}^1(3), \lambda_{i1}^2(3), \dots, \lambda_{i1}^r(3)\}$	$\lambda_{i2}(3) = \{\lambda_{i2}^1(3), \lambda_{i2}^2(3), \dots, \lambda_{i2}^r(3)\}$

È utilizzato un vettore etichetta temporaneo  $\eta_i(t) = \{\eta_i^k(t)\}$ ,  $k \in R$ . Per valutare se un ipercammino nuovo è dominato o no, si confronta il vettore etichetta temporaneo con le etichette dell'ipercammino Pareto-ottimale dal nodo  $i$  e al tempo  $t$ : se l'ipercammino temporaneo è dominato, viene scartato. Allo stesso modo, se domina uno o più degli ipercammini esistenti, questo viene mantenuto e vengono scartati quelli dominati.

In modo simile all'algoritmo del limite atteso più basso, l'algoritmo APS procede in via iterativa analizzando un nodo da una lista di nodi analizzabili, lavorando al contrario, cioè partendo dal nodo destinazione. L'algoritmo ELB costruisce un ipercammino da ogni nodo per ogni istante di partenza attraverso gli attuali sottoipercammini associati con i possibili tempi di arrivo al nodo successore. Questi ipercammini vengono analizzati per determinare se sono dominati, nel qual caso vengono scartati.

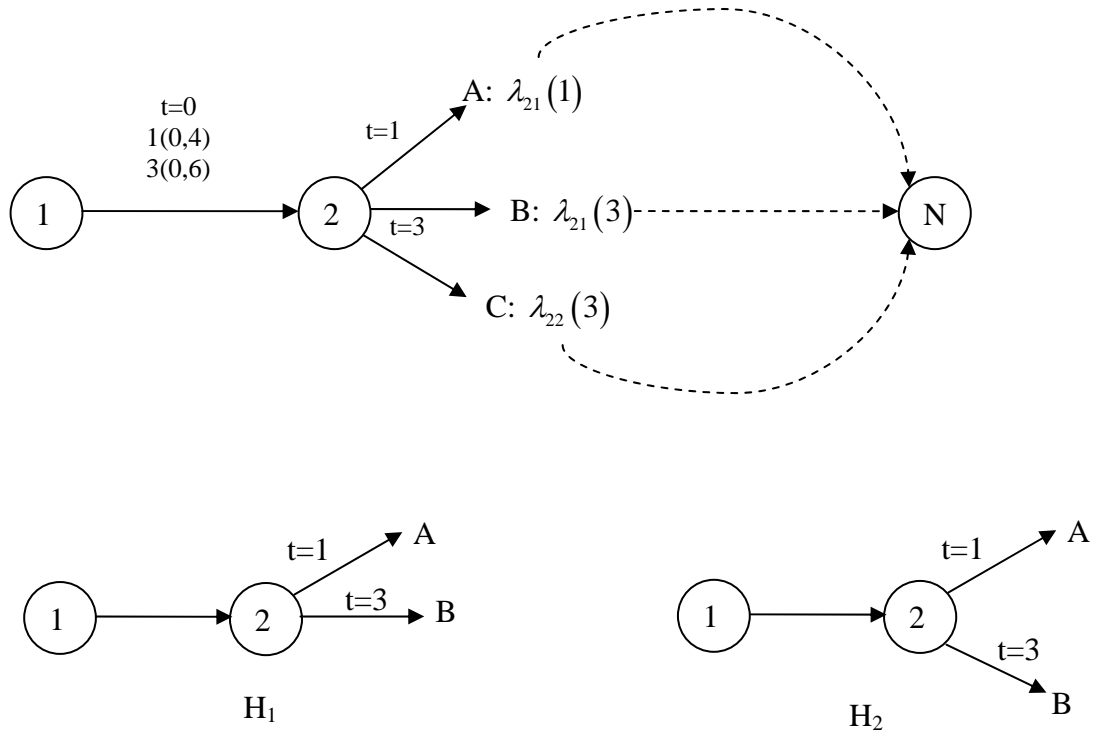
Per costruire un vettore etichetta associato ad un singolo ipercammino dal nodo  $i$  al tempo di partenza  $t$ , attraverso il nodo successore  $j$ , impiegando l'arco  $(i, j)$ , deve essere selezionato il vettore etichetta  $c_{ij}^{1z_1}(t)$  di un sottoipercammino  $\lambda_{jx}(t + c_{ij}^{1z_1}(t))$ ,  $x \in X_j(t + c_{ij}^{1z_1}(t))$  per ogni valore di  $z_1 \in \{1, 2, \dots, D\}$ , in funzione cioè dell'istante di percorrenza dell'arco  $(i, j)$ .

Ogni combinazione di  $z_1$  e  $\lambda_{jx}(t + c_{ij}^{1z_1}(t))$  è attribuita dalla coppia  $(z_1, x)$ .

L'ipercammino è costruito da queste  $D$  coppie (una per ogni possibile tempo di percorrenza per l'arco  $(i, j)$ ). Poiché possono esserci più coppie di  $(z_1, x)$  quando  $|X_j(t + c_{ij}^{1z_1}(t))| > 1$  per almeno un possibile tempo di arrivo al nodo  $j$ , può

essere possibile costruire più di un ipercammino dal nodo  $i$ . Infatti, ci sono esattamente  $\prod_{z=1}^D |X_j(t + c_{ij}^{z_1}(t))|$  ipercammini che possono essere costruiti.

Di seguito si fornisce un esempio in cui si illustra la costruzione dell'ipercammino:



La figura descrive un insieme di possibili ipercammini che collegano l'origine (il nodo 1) con la destinazione (il nodo  $N$ ) in una rete MSTV. Ci sono due possibili tempi di percorrenza per l'arco (1,2) al tempo  $t = 0$ : 1 con probabilità 0,4 e 3 con probabilità 0,6, che portano ai due corrispondenti tempi di arrivo al nodo 2, cioè  $t = 1$  e  $t = 3$ . Supponiamo che al nodo 2 esista un ipercammino A per l'istante  $t = 1$  con associata un'etichetta  $\lambda_{21}(1)$  e due ipercammini Pareto-ottimali B e C per l'istante  $t = 3$  con associate etichette rispettivamente  $\lambda_{21}(3)$  e  $\lambda_{22}(3)$ . Cioè  $X_2(1) = \{1\}$  e  $X_2(3) = \{1, 2\}$ . Quindi, al nodo 1, esistono due diverse combinazioni di coppie di  $(z_1, x)$ . Ciò risulta dalla generazione dei due ipercammini al nodo 1 per l'istante di partenza  $t = 0$ , cioè:

$$H_1 : (1,1), (2,1) \text{ e } H_2 : (1,1), (2,2).$$



Sono due le combinazioni che contribuiscono al calcolo del tempo medio di percorrenza di  $H_2$ : il primo possibile tempo di percorrenza dell'arco (1,2) e la prima etichetta al nodo 2 al tempo di arrivo corrispondente a questo tempo di percorrenza, ed il secondo possibile tempo di percorrenza e la seconda etichetta al nodo 2.

Sia  $Q$  l'insieme di queste coppie  $(z_1, x)$  che comprendono un singolo ipercammino  $H$ , cioè per  $H_2$ ,  $Q = \{(1,1), (2,2)\}$ . Il tempo medio di percorrenza per  $H_2$  può essere calcolato come segue:

$$\begin{aligned} \eta_1^1(0) &= \sum_{(z_1, x) \in Q} \left[ c_{12}^{1z_1}(0) + \lambda_{2x}^1(0 + c_{12}^{1z_1}(0)) \right] \cdot \rho_{12}^{1z_1}(0) = \\ &= \left[ c_{12}^{11}(0) + \lambda_{21}^1(0 + c_{12}^{11}(0)) \right] \cdot \rho_{12}^{11}(0) + \left[ c_{12}^{12}(0) + \lambda_{22}^1(0 + c_{12}^{12}(0)) \right] \cdot \rho_{12}^{12}(0) \end{aligned}$$

Notiamo come per questo calcolo siano stati usati due sottoipercammini, uno con istante di partenza  $t = 1$  e l'altro con  $t = 2$ .

Per consentire la ricostruzione efficiente degli ipercammini risultanti dopo il termine dell'algoritmo, sono utilizzati due cursori di percorso per ogni nodo  $i$  ad ogni istante  $t$  lungo l'ipercammino  $x$ . Questi cursori indicano il nodo successore e lo specifico sottoipercammino per il nodo successore per ogni possibile istante di partenza. Per ogni vettore etichetta  $x \in X_i(t)$ ,  $\pi_{ix}(t)$  indica il nodo successore che deve essere preso dal nodo  $i$  all'istante di partenza  $t$  lungo l'ipercammino  $x$ .

A differenza dell'algoritmo ELB, in cui è richiesto un solo cursore per questo scopo, per risolvere il problema multicriterio deve esserne mantenuto un insieme  $\left\{ q_{ix}(t) = \left\{ (z_1, x) \right\}_{con\ z_1 = \{1, 2, \dots, D\}} \right\}$  al fine di definire il sottoipercammino appropriato al nodo successore  $\pi_{ix}(t)$  per ogni valore di  $z_1$ , cioè per un particolare tempo di arrivo al nodo successore.

Prima che un vettore etichetta temporaneo sia controllato per trovare un'eventuale dominazione,  $\pi_{i_0}(t)$  e  $q_{i_0}(t)$  sono usati per mantenere temporaneamente l'informazione sul cammino di quel vettore etichetta.

Nell'esempio proposto in figura,  $\pi_{i_0}(0) = 2$  e  $q_{i_0}(0) = \{(1,1), (2,2)\}$ .

### I passi dell'algoritmo:

#### Passo 1 (Inizializzazione):

Inizializzare gli elementi delle etichette vettore ed i cursori di cammino.

- $\lambda_{i_1}^k(t) = \infty, \forall i \in \mathcal{V} / N, k \in \mathcal{R}, t \in S.$
- $\pi_{i_1}(t) = \infty, \forall i \in \mathcal{V}, t \in S.$
- $q_{i_1}(t) = \emptyset, \forall i \in \mathcal{V}, t \in S.$
- $X_i(t) = \{1\}, \forall i \in \mathcal{V}, t \in S.$
- $\lambda_{N_1}^k(t) = 0, \forall k \in \mathcal{R}, t \in S.$

Creare la lista dei nodi analizzabili e inserire il nodo destinazione  $N$  nella lista.

#### Passo 2 (Selezione del nodo da analizzare):

Se la lista dei nodi analizzabili è vuota, si va al passo 4; altrimenti, scegliere e cancellare un nodo dalla lista. Questo nodo diventa il corrente nodo  $j$ .

#### Passo 3 (Aggiornamento delle etichette dei nodi):

Per ogni  $i \in \Gamma^{-1}(j),$

per ogni  $t \in S,$

identificare  $\{c_{ij}^{1z_1}(t)\}$  e tutte le possibili  $Q$  combinazioni di  $\{(z_1, x)\}_{con\ z_1=\{1,2,\dots,D\}}$ .

Per ogni combinazione  $Q,$  calcolare il valore dell'etichetta temporanea  $\eta_i(t)$  come segue:

Per il tempo di percorrenza ( $k=1$ ):  $\eta_i^1(t) = \sum_{(z_1, x) \in Q} \left[ c_{ij}^{1z_1}(t) + \lambda_{jx}^1(t + c_{ij}^{1z_1}(t)) \right] \rho_{ij}^{1z_1}(t)$

Per altri criteri ( $k \neq 1$ ):  $\eta_i^k(t) = \sum_{(z_1, x) \in Q} \sum_{z_k=1}^D [c_{ij}^{kz_k}(t) + \lambda_{jx}^k(t + c_{ij}^{1z_1}(t))] \cdot \rho_{ij}^{1z_1}(t) \rho_{ij}^{kz_k}(t)$ .

Porre  $\pi_{i_0}(t) = j$  e  $q_{i_0}(t) = Q$ .

**Passo 4** (Controllo di dominio):

Controllare se l'ipercammino appena generato con associata l'etichetta  $\eta_i(t)$  è dominato attraverso paragoni a due a due con tutte le altre etichette non dominate in  $X_i(t)$ : se è dominato, deve essere scartato, altrimenti, aggiungere questa etichetta a  $X_i(t)$  ed eliminare tutte le etichette di ipercammini dominati da  $X_i(t)$ .

$$SE = SE \cup \{i\}$$

Tornare al passo 2.

**Passo 5** (Conclusione):

L'algoritmo termina con tutti i cammini Pareto-ottimali nel rispetto del valore atteso di ogni criterio da ogni origine al nodo destinazione  $N$ , per ogni istante di partenza  $t \in S$ .

### 2.3 Considerazioni sull'algoritmo APS

**I**n reti STV per problemi di cammino minimo a priori e con un singolo criterio, tutti i sottocammini di un cammino non dominato (nel rispetto di una varietà di criteri di dominio) con lo stesso nodo destinazione del cammino stesso, devono essere essi stessi non dominati. Estendendo questo concetto all'analisi multicriterio su reti STV:

*Lemma 1:* Dato un istante di partenza  $t \in S$ , qualsiasi ipercammino che contiene un sottoipercammino per lo stesso nodo destinazione dominato sarà dominato esso stesso.

*Discussione:* Può essere facilmente fornita una prova per esempio contrario. Si può mostrare che se un ipercammino contiene un sottoipercammino dominato, esiste un altro ipercammino che lo domina. Questa dimostrazione dettagliata è stata sviluppata da Opananon, 2004.

*Proposizione 6:* Al termine del processo, l'algoritmo genera tutti e soli gli ipercammini soluzione Pareto-ottimali.

*Prova:* Cominciamo mostrando che ogni etichetta finale generata dall'algoritmo è Pareto-ottimale. Sia  $\lambda_{ix}(t)$  un'etichetta associata ad uno degli ipercammini Pareto-ottimali per l'istante di partenza  $t$  dal nodo  $i$  determinata dall'algoritmo. Non può esistere nessun altro ipercammino  $v \notin X_i(t)$  con etichetta tale per cui:

$$\lambda_{iv}^k(t) \leq \lambda_{ix}^k(t) \quad \forall k \in \{1, 2, \dots, r\} \quad \text{e} \quad \lambda_{iv}^h(t) < \lambda_{ix}^h(t) \quad \text{per almeno un } h \in \{1, 2, \dots, r\}.$$

Supponiamo per assurdo che esista un ipercammino che soddisfi questa condizione.; quindi, una delle seguenti affermazioni deve essere vera:  $\lambda_{iv}(t)$  era dominata da un'altra etichetta o  $\lambda_{iv}(t)$  non è mai stata costruita nel passo 3 dell'algoritmo. Se esistesse un ipercammino che domina l'ipercammino a cui è associata l'etichetta  $\lambda_{iv}(t)$ , questo dominerebbe anche l'ipercammino cui è associata l'etichetta  $\lambda_{ix}(t)$ , contraddicendo l'assunzione che l'ipercammino cui è associata l'etichetta  $\lambda_{ix}(t)$  è Pareto-ottimale. Quindi  $\lambda_{iv}(t)$  non doveva essere costruito. Se  $\lambda_{iv}(t)$  non è mai stato costruito, o contiene un sottoipercammino dominato, per il lemma 1 è dominato, altrimenti la lista dei nodi analizzabili non potrebbe essere vuota, contraddicendo l'ipotesi di aver terminato il processo. Ciò stabilisce che tutti gli ipercammini soluzione presenti nella soluzione finale sono Pareto-ottimali. Ora è necessario provare che siano stati generati tutti gli ipercammini Pareto-ottimali.

Assumiamo che esista un ipercammino per alcuni istanti di partenza che non sia dominato da nessun altro ipercammino, ma questo non sia presente nell'insieme finale delle soluzioni.

Questo ipercammino potrebbe essere escluso dall'insieme finale delle soluzioni solo se non fosse mai stato creato. Cioè, un sottoipercammino di questo ipercammino deve essere dominato o un sottoipercammino di questo cammino non è mai stato generato. Nel primo caso, se il sottoipercammino è dominato, per il lemma 1 ogni ipercammino che contenga questo sottoipercammino deve essere dominato. Il secondo caso può accadere solo se il sottoipercammino contiene a sua volta un sottoipercammino dominato, quindi sarebbe dominato, o se non c'è nessun cammino tra l'origine del sottoipercammino e il nodo destinazione, quindi l'ipotetico ipercammino non potrebbe esistere. Quindi, nessun cammino escluso dall'insieme delle soluzioni finali può essere Pareto-ottimale.

*Proposizione 7:* L'algoritmo APS ha, nel peggiore dei casi, complessità computazionale esponenziale.

*Discussione:* Come affermato precedentemente, è possibile che nel peggiore dei casi tutti i possibili ipercammini siano Pareto-ottimali. Di conseguenza, l'algoritmo APS, che cerca tutte le soluzioni Pareto-ottimali, è esponenziale nel peggior caso di complessità computazionale.

## 2.4 L'algoritmo ALKDS I

l'algoritmo APS genera tutti gli ipercammini Pareto-ottimali. Il lato positivo di questo algoritmo è che un decisore potrebbe a posteriori scegliere una singola soluzione migliore fra tutti gli ipercammini Pareto-ottimali, ma purtroppo la generazione di tutti questi ipercammini può richiedere enormi sforzi computazionali. Per questo motivo, i due ricercatori si sono posti il problema di presentare un algoritmo per generare efficientemente un singolo ipercammino ottimo rappresentando la struttura di preferenze del decisore attraverso una funzione di disutilità. Invece che costruire diversi vettori etichetta per tutti gli ipercammini Pareto-ottimali, quest'algoritmo mantiene solo un vettore etichetta  $\lambda_i(t) = \{\lambda_i^k(t)\}$  con  $k \in \mathcal{R}$  per

ogni nodo  $i \in \mathcal{V}$  e per ogni istante di partenza  $t \in \mathcal{S}$  dove  $\lambda_i^k(t)$  rappresenta il valore atteso per l'attributo del cammino rispetto al criterio  $k$ . Per ogni  $i \in \mathcal{V}$  e  $t \in \mathcal{S}$  viene utilizzata un'etichetta  $U_i(t)$ . Assumiamo che la funzione di disutilità  $U_i(t)$  sia lineare e possa essere scritta come segue:

$$U_i(t) = \sum_{k=1}^r w^k \cdot \lambda_i^k(t) \quad \text{Equazione 1}$$

dove  $w^k$  è il peso per il criterio  $k$  basato sulle preferenze del viaggiatore e  $\sum_{k=1}^r w^k = 1$ .

Prima della conclusione,  $U_i(t)$  fornisce un limite superiore alla disutilità attesa per viaggiare all'istante  $t$  dal nodo  $i$  a destinazione. Terminato il processo, mantiene la minore disutilità attesa di tutti gli ipercammini dal nodo  $i$  all'istante di partenza  $t$ .

Come l'algoritmo APS, questo algoritmo è un'estensione dell'algoritmo ELB. Ad ogni nodo vengono calcolati i valori attesi per ogni criterio e la disutilità è calcolata utilizzando l'equazione 1 sopra citata. Le etichette temporanee  $\eta_i^k(t)$  e  $v_i(t)$  mantengono rispettivamente i valori  $\lambda_i^k(t)$  e  $U_i(t)$  prima dell'aggiornamento. Se  $v_i(t) < U_i(t)$ , allora  $U_i(t)$  viene posto uguale a  $v_i(t)$  e i  $\lambda_i^k(t) \forall k \in \mathcal{R}$  sono aggiornati di conseguenza. Al termine del processo,  $\{\lambda_i^k(t)\}_{con\ k \in \mathcal{R}}$  contiene i valori attesi dell'ipercammino per ogni criterio con associata la minore disutilità attesa  $U_i(t)$  all'istante di partenza  $t$  dal nodo  $i$  alla destinazione  $N$ .

Per costruire gli ipercammini con la minore disutilità attesa efficientemente, è richiesto un singolo cursore,  $\pi_i(t)$ , che specifichi il nodo successore per viaggiare dal nodo  $i$  all'istante di partenza  $t$ .

## I passi dell'algoritmo:

### Passo 1 (Inizializzazione):

Inizializzare gli elementi delle etichette vettore e dei cursori.

- $\lambda_i^k(t) = \infty, \forall i \in \mathcal{V} / N, k \in \mathcal{R}, t \in \mathcal{S}.$
- $U_i(t) = \infty, \forall i \in \mathcal{V} / N, t \in \mathcal{S}.$
- $\pi_i(t) = \infty, \forall i \in \mathcal{V}, t \in \mathcal{S}.$
- $\lambda_N^k(t) = 0, \forall k \in \mathcal{R}, t \in \mathcal{S}.$
- $U_N(t) = 0, \forall t \in \mathcal{S}.$

Creare la lista dei nodi analizzabili e inserire il nodo  $N$  nella lista.

### Passo 2 (Selezione del nodo da analizzare):

Se la lista dei nodi analizzabili è vuota, si va al passo 4; altrimenti scegliere e cancellare un nodo dalla lista. Questo nodo diventa il corrente nodo  $j$ .

### Passo 3 (Aggiornamento delle etichette dei nodi):

Per ogni  $i \in \Gamma^{-1}(j)$ ,

per ogni  $t \in \mathcal{S}$ ,

$$v_i(t) = \sum_{k=1}^r w^k \cdot \eta_i^k(t), \text{ dove}$$

$$k=1: \eta_i^1(t) = \sum_{z_1=1}^D \left[ c_{ij}^{1z_1}(t) + \lambda_j^1(t + c_{ij}^{1z_1}(t)) \right] \cdot \rho_{ij}^{1z_1}(t)$$

$$k \neq 1: \eta_i^k(t) = \sum_{z_1=1}^D \sum_{z_k=1}^D \left[ c_{ij}^{kz_k}(t) + \lambda_j^k(t + c_{ij}^{1z_1}(t)) \right] \rho_{ij}^{1z_1}(t) \rho_{ij}^{kz_k}(t).$$

Se  $v_i(t) < U_i(t)$ , allora  $U_i(t) = v_i(t)$ ,  $\pi_i(t) = j$ ,  $\lambda_i^k(t) = \eta_i^k(t) \forall k \in \mathcal{R}$  e

$$SE = SE \cup \{1\}.$$

Tornare al passo 2.

#### **Passo 4 (Conclusione):**

L'algoritmo termina con gli ipercammini a disutilità attesa minima per ogni istante di partenza  $t \in S$ , da ogni origine al nodo destinazione  $N$  per un dato insieme di  $k$  criteri di peso  $w^k$  ed una funzione lineare di disutilità.

### *2.5 Considerazioni sull'algoritmo ALEDS I*

**R**

itenendo utile affrontare, congiuntamente all'analisi degli algoritmi, un'analisi di efficienza computazionale, i ricercatori hanno enunciato la seguente proposizione con la quale si stabilisce il limite massimo di complessità.

*Proposizione 8:* La complessità computazionale di questo algoritmo, con sistema FIFO nella selezione dei nodi è, nel peggior caso,  $\sim O(V^3 \cdot I^2 \cdot P^2 \cdot R)$ , dove  $V = |V|$  è il numero dei nodi della rete,  $I$  è il numero degli intervalli di tempo nel periodo di interesse,  $P = |P|$  è il numero massimo di possibili pesi degli archi per un dato criterio e  $R = |R|$  è il numero dei criteri.

*Discussione:* La complessità di questo algoritmo può essere ricavata in maniera simile a come fatto per l'algoritmo ELB. La sola differenza è che il passo 3 dell'algoritmo ELB ha una complessità computazionale  $\sim O(V \cdot I \cdot P)$ , mentre il passo 3 dell'algoritmo ALEDS I ha complessità  $\sim O(V \cdot I \cdot P^2 \cdot R)$ .

È importante notare come la complessità del passo 3 dell'algoritmo ALEDS I sia maggiore di quella dell'algoritmo ELB, in senso assoluto, solo per il numero dei pesi degli archi; come intuibile, la complessità è proporzionale anche per ogni criterio, in quanto equivale a risolvere  $R$  volte il problema.



## 2.6 L'algoritmo ALEDS II

Questo algoritmo è un perfezionamento dell'algoritmo ALEDS I. Nell'algoritmo precedente, il valore atteso per ogni criterio era calcolato prima di calcolare la disutilità per l'ipercammino associato. Tutte le etichette di cammino che risultavano dovevano essere mantenute. Invece, l'algoritmo ALEDS II valuta la disutilità attesa direttamente e mantiene solo il valore minimo per ogni nodo ed istante di partenza; non si presenta quindi il bisogno di mantenere le etichette individuali  $\lambda_i(t) = \left\{ \lambda_i^k(t) \right\}_{con\ k \in \mathcal{R}}$ ,  $\forall i \in \mathcal{V}, t \in \mathcal{S}$  richieste nella prima variante dell'algoritmo.

Mentre l'algoritmo ALEDS I è più intuitivo, attraverso le modifiche utilizzate in questa seconda variante, si raggiungono significativi risparmi negli sforzi computazionali e nei requisiti di memoria.

*Lemma 2:* Il calcolo di  $v_i(t)$  nel passo 3 dell'algoritmo ALEDS I può essere svolto attraverso la seguente equazione:

$$v_i(t) = \sum_{k=1}^r w^k \cdot \left( \sum_{z_k=1}^D c_{ij}^{kz_k}(t) \cdot \rho_{ij}^{kz_k}(t) \right) + \sum_{z_1=1}^D U_j(t + c_{ij}^{1z_1}(t)) \cdot \rho_{ij}^{1z_1}(t)$$

*Prova:*  $v_i(t) = \sum_{k=1}^r w^k \cdot \eta_i^k(t)$  (il calcolo di  $\eta_i^k(t)$ ,  $\forall k \in \mathcal{R}$  è descritto nel passo 3 dell'algoritmo)

$$v_i(t) = w^1 \sum_{z_1=1}^D \left[ c_{ij}^{1z_1}(t) + \lambda_j^1(t + c_{ij}^{1z_1}(t)) \right] \rho_{ij}^{1z_1}(t) + \\ + \sum_{k=2}^r w^k \left[ \sum_{z_1=1}^D \sum_{z_k=1}^D \left[ c_{ij}^{kz_k}(t) + \lambda_j^k(t + c_{ij}^{1z_1}(t)) \right] \rho_{ij}^{1z_1}(t) \rho_{ij}^{kz_k}(t) \right] =$$

È stato estratto il criterio del tempo.

$$= w^1 \sum_{z_1=1}^D c_{ij}^{1z_1}(t) \rho_{ij}^{1z_1}(t) + \sum_{k=2}^r w^k \left( \sum_{z_1=1}^D \sum_{z_k=1}^D c_{ij}^{kz_k}(t) \rho_{ij}^{1z_1}(t) \rho_{ij}^{kz_k}(t) \right) + \\ + w^1 \sum_{z_1=1}^D \lambda_j^1(t + c_{ij}^{1z_1}(t)) \rho_{ij}^{1z_1}(t) + \sum_{k=2}^r w^k \left( \sum_{z_1=1}^D \sum_{z_k=1}^D \lambda_j^k(t + c_{ij}^{1z_1}(t)) \rho_{ij}^{1z_1}(t) \rho_{ij}^{kz_k}(t) \right) =$$

È stato svolto il prodotto interno alle sommatorie le quali sono state scisse in due sommatorie.

$$= w^1 \sum_{z_1=1}^D c_{ij}^{1z_1}(t) \rho_{ij}^{1z_1}(t) + \sum_{k=2}^r w^k \left( \sum_{z_k=1}^D c_{ij}^{kz_k}(t) \rho_{ij}^{kz_k}(t) \right) +$$

$$+ w^1 \sum_{z_1=1}^D \lambda_j^1(t + c_{ij}^{1z_1}(t)) \rho_{ij}^{1z_1}(t) + \sum_{k=2}^r w^k \left( \sum_{z_1=1}^D \lambda_j^k(t + c_{ij}^{1z_1}(t)) \rho_{ij}^{1z_1}(t) \right) =$$

Poiché  $\sum_{z_1=1}^D \rho_{ij}^{1z_1}(t) = 1$  e  $\sum_{z_k=1}^D \rho_{ij}^{kz_k}(t) = 1$  le doppie sommatorie si semplificano.

$$= \sum_{k=1}^r w^k \cdot \left( \sum_{z_k=1}^D c_{ij}^{kz_k}(t) \cdot \rho_{ij}^{kz_k}(t) \right) + \sum_{z_1=1}^D U_j(t + c_{ij}^{1z_1}(t)) \cdot \rho_{ij}^{1z_1}(t)$$

Sono state ricomposte le sommatorie, includendo anche il primo elemento, dal momento in cui il loro contenuto è identico.

Il Lemma 2 mostra come l'algorithmo ALEDS II possa essere semplificato attraverso un calcolo più efficiente del valore atteso di disutilità. I due algoritmi producono soluzioni identiche.

### I passi dell'algorithmo:

#### Passo 1 (Inizializzazione):

Inizializzare le etichette ed i cursori.

- $U_i(t) = \infty, \forall i \in \mathcal{V} / N, t \in \mathcal{S}$
- $\pi_i(t) = \infty, \forall i \in \mathcal{V}, t \in \mathcal{S}$
- $U_N(t) = 0, \forall t \in \mathcal{S}$

Creare la lista dei nodi analizzabili e inserire il nodo destinazione nella lista.

#### Passo 2 (Selezione del nodo da analizzare):

Se la lista dei nodi analizzabili è vuota, si va al passo 4; altrimenti scegliere e cancellare un nodo dalla lista. Questo nodo diventa il corrente nodo  $j$ .

**Passo 3** (Aggiornamento delle etichette dei nodi):

Per ogni  $i \in \Gamma^{-1}(j)$ ,

per ogni  $t \in S$ ,

$$v_i(t) = \sum_{k=1}^r w^k \cdot \left( \sum_{z_k=1}^D c_{ij}^{kz_k}(t) \cdot \rho_{ij}^{kz_k}(t) \right) + \sum_{z_1=1}^D U_j(t + c_{ij}^{1z_1}(t)) \cdot \rho_{ij}^{1z_1}(t).$$

Se  $v_i(t) < U_i(t)$ , allora  $U_i(t) = v_i(t)$  e  $\pi_i(t) = j$ .  $SE = SE \cup \{1\}$ .

Ritornare al passo 2.

**Passo 4** (Conclusione):

L'algoritmo termina con gli ipercammini a disutilità attesa minima per ogni istante di partenza  $t \in S$ , da ogni origine al nodo destinazione  $N$  per un dato insieme di  $k$  criteri di peso  $w^k$  ed una funzione lineare di disutilità.

## 2.7 Considerazioni sull'algoritmo ALLEDG II

**q** li autori sottolineano come sia significativo il fatto che entrambi gli algoritmientino sul fatto che un ipercammino a minima disutilità attesa contenga solo sottoipercammini con minima disutilità attesa verso la stessa destinazione, come stabilito nel lemma 3.

*Lemma 3:* Qualsiasi ipercammino a minima disutilità attesa contiene solo sottoipercammini a minima disutilità attesa verso la stessa destinazione.

*Discussione:* In maniera simile alla prova del lemma 1, la prova di questo lemma può essere fornita prontamente con un esempio contrario. La prova consiste nello stabilire che, se un ipercammino a disutilità attesa minima contenesse un sottoipercammino che non sia a disutilità attesa minima dal suo nodo origine, un cammino contenente un sottoipercammino a disutilità attesa minima da

quell'origine avrebbe una disutilità attesa minima inferiore. Dettagli di questa prova sono stato esposti da Opananon (2004).

*Proposizione 9:* Una volta terminati, i due algoritmi ALEDS I e II forniscono gli ipercammini a disutilità attesa minima per ogni nodo e per ogni istante di partenza nell'intervallo di tempo preso in considerazione.

La prova di questa proposizione è una diretta conseguenza della prova fornita per l'algoritmo ELB in Miller-Hooks e Mahamassani (2000).

*Proposizione 10:* La complessità computazionale nel caso peggiore per l'algoritmo ALEDS II con lista dei nodi analizzabili con logica FIFO è  $\sim O(V^3 \cdot I^2 \cdot P \cdot R)$ .

*Discussione:* La complessità dell'algoritmo ALEDS II è  $\sim O(P)$  migliore di quella dell'algoritmo ALEDS I. Questo perché il passo 3 ha complessità  $\sim O(V \cdot I \cdot P \cdot R)$ , mentre il passo 3 dell'algoritmo ALEDS I ha complessità  $\sim O(V \cdot I \cdot P^2 \cdot R)$ .

Notiamo come la complessità computazionale dell'algoritmo ALEDS II sia solo  $\sim O(R)$  peggiore di quella dell'algoritmo ELB che considera un solo criterio.

## 2.8 *Esperimenti su convenienti reti standard*

**I**n problemi collegati che possono essere ricondotti a procedure di correzione di etichette, in cui sono associate più vettori etichetta per ogni nodo, è talvolta vantaggioso includere coppie etichetta-nodo nella lista dei nodi analizzabili piuttosto che semplicemente i nodi. Mentre nell'algoritmo APS sono utilizzati diversi vettori etichetta, per via del modo in cui sono stati creati, non c'è beneficio ad includere la coppia etichetta-nodo nella lista dei nodi analizzabili, così che solo i nodi so-

no inseriti nella lista. Nell'analisi della complessità degli algoritmi ALEDS si assume che la lista dei nodi analizzabili abbia una struttura FIFO.

Comunque, è possibile impiegare una struttura differente, come perfezionamento della lista.

I due ricercatori hanno quindi effettuato una serie di esperimenti su reti generate casualmente per verificare le prestazioni degli algoritmi.

Il numero dei nodi, il numero degli intervalli di tempo, il numero degli elementi nelle funzioni di probabilità e il numero dei criteri sono predeterminati.

Conformemente alle reti di trasporto, le condizioni di ingresso e di uscita sono, in media, quattro e variano tra 2 e 9. Per ogni criterio viene scelta una distribuzione uniforme con un limite inferiore di un'unità. Il limite superiore è progettato per aumentare linearmente da 5 a 10 unità nella prima metà del periodo di punta e decrescere linearmente di 5 unità nella seconda metà del periodo di punta.

Gli algoritmi sono stati programmati in C++ e calcolati su una workstation professionale DEC Alpha XP1000 con 1 gigabyte di RAM e 2 gigabyte per lo swap, con sistema operativo Digital 4.0E.

Sono state effettuate tre serie di esperimenti. Sono state sperimentate le prestazioni medie in termini di tempo di esecuzione dell'algoritmo APS. Sono state considerate reti composte da 25 nodi, periodi di punta di 15 e 30 intervalli di tempo, 3 elementi in ogni funzione densità di probabilità e 2 criteri.

L'algoritmo APS ha ottenuto pessime prestazioni in questi esperimenti, in termini di tempo di esecuzione e requisiti di memoria, poiché un enorme numero di ipercammini è Pareto-ottimale, come già argomentato. Non sono riportati ulteriori dettagli riguardanti questi esperimenti.

Il risparmio medio in termini computazionali che può essere ottenuto usando l'algoritmo ALEDS II al posto dell'algoritmo ALEDS I è stato valutato attraverso esperimenti compiuti su 8 reti con 100 o 500 nodi, 60 o 120 intervalli di tempo, 5 elementi per ogni funzione densità di probabilità e 2 o 6 criteri. Ulteriori test dell'algoritmo ALEDS II sono stati condotti su reti composte da 50, 100, 500 e 1000 nodi, un periodo di punta di 15, 30, 60 e 120 intervalli di tempo, 5, 10 e 20

elementi per funzione densità di probabilità e 2, 4, 6 criteri. Sono state svolte 30 elaborazioni, corrispondenti a 30 destinazioni scelte casualmente, per ogni rete e configurazione degli intervalli di tempo nel periodo di punta, numero di elementi nella funzione densità di probabilità e numero di criteri. Sono riportate di seguito le medie di queste 30 elaborazioni.

Nella tabella seguente si può osservare il miglioramento ottenuto in termini computazionali nel passare dall'algorithmo ALEDS I alla variante ALEDS II.

R	ALEDS I				ALEDS II			
	V = 100		V = 500		V = 100		V = 500	
	I = 60	I = 120	I = 60	I = 120	I = 60	I = 120	I = 60	I = 120
2	0,074	0,157	0,333	0,68	0,035	0,072	0,195	0,393
6	0,153	0,304	0,784	1,557	0,056	0,112	0,292	0,574

La tabella mostra come l'algorithmo ALEDS II sia approssimativamente due volte più veloce dell'algorithmo ALEDS I per i problemi con 2 criteri e 3 volte più veloce per i problemi con 6 criteri.

In una rete con 25 nodi, 3 elementi per funzione densità di probabilità, 15 intervalli di tempo e due criteri, il numero medio e massimo di ipercammini Pareto-ottimali (come generati dall'algorithmo APS) sono rispettivamente 3,3 e 119. Si osserva quindi come gli algoritmi ALEDS possano offrire risparmi significativi sia in termini di tempo di calcolo, sia di requisiti di memoria rispetto alle procedure che generano tutti gli ipercammini Pareto-ottimali.

Le prestazioni computazionali dell'algorithmo ALEDS II sono state successivamente testate attraverso ulteriori esperimenti numerici. I risultati mostrano come l'algorithmo riporti ottime prestazioni, migliori di quanto atteso data la complessità, anche in reti con 1000 nodi.

Nonostante ciò, come previsto, esperimenti su reti dalle dimensioni troppo elevate non possono essere completati per via degli eccessivi requisiti di memoria. La nascita di questi requisiti è dovuta in larga parte all'utilizzo di vettori tetradimensionali utilizzati per gestire le etichette vettore. Una più efficiente programmazione dell'algorithmo può permettere la soluzione di problemi di queste dimensioni.

## 2.9 Conclusioni

**I**n questo lavoro, si è discusso il problema di generare ipercammini che cerchino di minimizzare il valore atteso di diversi criteri in reti MSTV. È stato proposto un algoritmo esatto, l'algoritmo APS, per generare tutti gli ipercammini Pareto-ottimali. Dati i diversi criteri, tali soluzioni permettono al conducente di scegliere in modo adattativo il cammino da seguire in ogni locazione intermedia tra tutte le strategie non dominate.

In quanto generazione di tutte le soluzioni Pareto-ottimali, l'algoritmo APS ha una complessità esponenziale. Infatti, è richiesto un algoritmo che fornisca una singola soluzione "miglior compromesso". Piuttosto che generare tutte le soluzioni Pareto-ottimali e scegliere a posteriori una singola soluzione, se le preferenze di un decisore possono essere rappresentate da una funzione di disutilità lineare, un approccio più diretto ed efficiente è fornito dagli algoritmi ALEDS I e II. Sebbene meno intuitivo del primo, l'algoritmo ALEDS II fornisce miglioramenti sostanziali nella complessità computazionale e nei requisiti di memoria.

Le strategie generate degli algoritmi proposti, consentono ai viaggiatori di scegliere i loro cammini in base alla ragionevole previsione delle condizioni del traffico. È data importanza allo scambio reciproco che avviene tra i diversi attributi durante il processo di selezione del cammino.

Problemi che richiedono la selezione di tali cammini si possono incontrare in una grande varietà di situazioni, tra cui la scelta del percorso per unità di emergenza (ambulanze, polizia, vigili del fuoco e altri), veicoli che trasportano materiali pericolosi e viaggiatori singoli in città con strade congestionate, così come la selezione del percorso di pacchetti di dati in reti telematiche.

Per quanto riguarda la complessità computazionale, i risultati degli esperimenti condotti hanno mostrato come, in termini di tempi medi di calcolo, l'algoritmo APS non sia efficiente in reti di dimensioni maggiori. Comunque, tali procedure

esatte possono risultare utili nel fornire soluzioni di parametro a piccoli problemi, ad esempio nello sviluppo di approcci più efficienti ma euristici.

I risultati degli esperimenti computazionali hanno anche mostrato che il meno intuitivo algoritmo ALEDS II superi nettamente il più intuitivo ALEDS I. In aggiunta, sembra che le prestazioni medie dell'algoritmo ALEDS II siano inferiori a quanto previsto dall'analisi della complessità computazionale nel peggior caso.



ORAPAMONO

RE

Riduzione del problema

# Capitolo 3

## Riduzione del problema

*Sommario: 3.1 La riduzione al problema a criterio singolo - 3.2 Il lavoro di P. Modesti – A. Sciomachen - 3.3 La misura di utilità proposta - 3.4 Il problema della scelta dei coefficienti - 3.5 Una possibile integrazione - 3.6 Conclusioni*

### *3.1 La riduzione al problema a criterio singolo*

**d**a quanto osservato nei precedenti capitoli, il problema nel suo complesso assume velocemente e facilmente dimensioni rilevanti. La relativa complessità computazionale ne comporta quindi la scarsa applicabilità a casi reali, benché il modello proposto da Opananon e Miller-Hooks, commentato nel precedente capitolo, possa già offrire risultati accettabili su reti di dimensioni medie.

Una possibile riduzione del problema è nel ridurre il numero di criteri attraverso un'omogeneizzazione delle unità di misura dei pesi degli archi.

Cioè, rendendo comparabili tra loro i pesi degli archi, si potrebbe giungere, attraverso la somma diretta dei pesi degli archi, ad un grafo per cui l'attributo di ogni singolo arco sia la somma di tutti gli attributi dell'arco stesso, ciascuno adattato secondo un coefficiente.

Svolta questa operazione, ci troveremmo di fronte ad un problema STV, per cui esistono in letteratura diversi algoritmi efficienti per la ricerca del cammino minimo.

In questo caso, la creazione di un singolo modello personalizzabile su misura delle preferenze individuali dei singoli utenti ridurrebbe il problema alla scelta dei coefficienti.

Un simile lavoro è stato proposto nel 1996 (pubblicato nel 1998 in *"European Journal of Operational Research"*) da Paola Modesti e Anna Sciomachen, attraverso un modello che affrontava il problema del cammino minimo in reti urbane multimodali, con l'obiettivo di massimizzare una funzione di utilità costituita da una funzione che riassume il costo generalizzato dato dalla percorrenza dell'arco.

Nel lavoro in questione, viene presentato un modello in cui le funzioni di utilità sono state sviluppate tenendo conto delle differenti preferenze dei soggetti.

Inoltre, l'approccio proposto è stato sviluppato per analizzare la rete di trasporto urbano della città di Genova, e nell'articolo citato sono forniti i primi dati sperimentali.

### *3.2 Il lavoro di Modesti – Sciomachen*

**I**l lavoro presenta un primo paragrafo di introduzione al problema, in cui viene descritto sinteticamente il contenuto dell'articolo; nel secondo paragrafo passa alla descrizione analitica del tipo di rete considerato, fornendo le definizioni del problema.

La parte maggiormente interessante per lo sviluppo del presente lavoro è quella contenuta nei paragrafi 3 e 5, in cui vengono proposte e descritte analiticamente le funzioni di utilità. Nei paragrafi successivi si passa poi alla descrizione e all'analisi del caso sperimentale concreto applicato alla rete urbana di Genova.

La rete definita dal lavoro è basata sul concetto di sottografo, poiché si analizza anche il problema della multimodalità.

Vengono infatti considerate tre distinte modalità di trasporto:

- ⊗ Modalità privata: in questa modalità rientrano gli spostamenti effettuati con veicoli privati (automobile, motociclo).

- ⊗ Modalità pubblica: in questa modalità rientrano gli spostamenti effettuati con mezzi pubblici (autobus, metropolitana, treno).
- ⊗ Modalità pedonale: gli spostamenti vengono effettuati a piedi.

Il grafo viene definito come l'unione dei tre sottografi rappresentanti, ciascuno, la rete della rispettiva modalità di trasporto e tali per cui l'intersezione dei grafi, eventualmente presi a due a due, non sia l'insieme vuoto; quest'ipotesi è necessaria per permettere il cambiamento da una modalità ad un'altra. I nodi dei sottografi appartenenti all'insieme intersezione sono collegati tra loro attraverso archi il cui peso rappresenta il costo per passare da una modalità all'altra.

Ad ogni arco sono attribuiti due pesi, il primo dei quali rappresenta il tempo impiegato per percorrerlo, mentre il secondo rappresenta il costo.

Vengono definiti **centroidi** i nodi che rappresentano luoghi particolarmente rilevanti nell'area urbana.

Sono stabilite inoltre particolari regole di transizione da una modalità ad un'altra:

- ⊗ Nessun arco di transizione permette di spostarsi dalla modalità pubblica o pedonale a quella privata, poiché si suppone che, percorso un tratto senza il proprio veicolo, non sia più possibile utilizzarlo nello stesso itinerario per giungere a destinazione.
- ⊗ Non è possibile passare direttamente dal grafo della modalità privata a quello della modalità pubblica, poiché si suppone che dal luogo di parcheggio del veicolo alla fermata dell'autobus si giunga attraverso la modalità pedonale.
- ⊗ Il peso dell'arco di transizione dalla modalità privata a quella pubblica è scomposto in due: una componente relativa al costo e al tempo necessari per trovare un parcheggio ed una componente relativa al costo e al tempo necessari per spostarsi dal parcheggio al nodo della rete pubblica.

### 3.3 La misura di utilità proposta

**A**ssumiamo che l'utente abbia una funzione di utilità che desidera massimizzare. Attraverso questa ipotesi, è possibile rendere omogenee le differenti variabili  $x_k$  con  $k=1, \dots, p$  che concorrono a determinare la scelta dell'utente. Come peso degli archi andremo quindi a considerare una funzione di utilità reale dipendente dalle variabili  $x_1, \dots, x_p$ .

Per ogni cammino ammissibile  $\Pi$  da un certo nodo origine ad un nodo destinazione assumiamo che sia  $\mathcal{V}(\Pi) = \mathcal{V}_{\Pi_n}(x_1, x_2, \dots, x_p)$  la variabile aleatoria per ogni utente e che possa essere rappresentata come  $\mathcal{V} = U + \varepsilon$ . Operiamo questa distinzione assumendo che  $U$  sia una funzione reale definita nell'insieme dei pesi dei cammini, costante per tutti gli utenti, chiamata *utilità rappresentativa* o *utilità sistematica*, ed  $\varepsilon$  sia un *residuo aleatorio* caratteristico per ogni singolo utente.

Questa scelta è stata attuata poiché, anche se la funzione è associata individualmente a ciascun utente, come è ragionevole ritenere, alcune componenti assumono la stessa importanza per tutti gli utenti.

Dal punto di vista della teoria delle decisioni, il problema può essere visto come segue: ogni decisore ha una propria funzione di utilità che dovrebbe essere, escludendo possibili errori di percezione, interamente deterministica. L'analista che desidera massimizzare le funzioni di utilità di tutti i decisori, si troverà a dover gestire una enorme (ed inutile) quantità di soluzioni Pareto-ottimali.

Quindi, al fine di costruire modelli efficienti dal punto di vista computazionale, la funzione di utilità per ogni cammino è scomposta nella somma di una componente costante ( $U$ ) che rappresenta l'utilità che gli utenti percepiscono similmente ed una componente ( $\varepsilon$ ), diversa per ogni utente, aleatoria per l'analista, il quale è in grado solo di attribuire una funzione di distribuzione ricavata attraverso la raccolta e l'analisi di dati relativi al comportamento dei singoli utenti.

Quindi il problema diventa una previsione il cui obiettivo non è trovare il cammino con l'utilità maggiore per tutti gli individui, ma individuare il cammino che dovrebbe essere scelto con maggiore probabilità.

Gli autori proseguono ora proponendo una funzione di utilità sistematica, rimandando ad un lavoro successivo lo studio del problema relativo al residuo aleatorio.

Generalmente si assume che la funzione di utilità sistematica sia una funzione lineare dei suoi attributi, cioè  $U(x_1, x_2, \dots, x_p) = \sum_{i=1}^p \beta_i x_i$  dove  $x_k$  è il valore del k-esimo attributo e  $\beta_k$  è il coefficiente di  $x_k$ .

Gli autori operano una distinzione tra le diverse variabili, considerando sei attributi relativi al trasporto stesso ed uno relativo al contesto economico-sociale.

Per quanto riguarda la durata del cammino, le variabili considerate sono:

- ⊗ Il tempo speso nella modalità privata, suddiviso a sua volta, come già specificato precedentemente, nella somma tra il tempo speso alla guida del veicolo, il tempo impiegato per trovare un parcheggio libero ed il tempo per spostarsi a piedi dal parcheggio al nodo.
- ⊗ Il tempo speso su un mezzo pubblico.
- ⊗ Il tempo di attesa del mezzo pubblico.
- ⊗ Il tempo trascorso camminando.

Per quanto riguarda il costo del cammino, le variabili considerate sono 3, ma trattandosi di un'unità di misura omogenea di per sé, il denaro, possono essere agevolmente sintetizzate da un'unica variabile costo data dalla somma delle tre:

- ⊗ Il costo della modalità privata, dato da un costo variabile rappresentativo del carburante più una somma relativa al costo del parcheggio.
- ⊗ Il costo della rete di trasporto pubblico.
- ⊗ Il costo della modalità pedonale (che nel lavoro è considerato nullo).

Sembra opportuno non effettuare la stessa sintesi per le variabili temporali, poiché il tempo speso nel trasporto è percepito in modo diverso dai diversi utenti; per questo motivo, si rende necessario l'uso di uno specifico coefficiente per ponderare la misura a seconda dell'utente.

Per pesare ogni variabile, si ricorre ad un coefficiente per ogni variabile; poiché un aumento del valore della variabile comporta una diminuzione dell'utilità dell'utente, i coefficienti  $\beta_k$  dovranno essere negativi. Considerando, al contrario, i coefficienti positivi, il problema può essere visto come la minimizzazione di una funzione di disutilità.

### *3.4 Il problema della scelta dei coefficienti*

**n**el paragrafo 5, il lavoro proposto da P. Modesti e A. Sciomachen offre un modello puramente esemplificativo di possibili valori dei coefficienti degli attributi, rimandando ad un lavoro futuro lo sviluppo di un modello per una stima di questi coefficienti attraverso ricerche su preferenze rivelate o preferenze determinate, al fine di poter applicare il metodo della massima verosimiglianza.

Attraverso questa operazione, si è quindi ridotto il problema multicriterio al problema della stima dei coefficienti.

Il problema in questione si presenta anche nel lavoro presentato da Opananon e Miller-Hooks, rappresentando tuttavia, in quel caso, un fattore di rilevanza ridotta. L'analisi dei coefficienti assume infatti un rilievo maggiore nel caso di reti multimodali, laddove si presenta il problema di valutare in maniera differente anche un criterio apparentemente semplice ed uniforme quale il tempo.

In merito, i due ricercatori italiani hanno distinto infatti il tempo speso tra le diverse modalità, ritenendo che il tempo trascorso camminando comporti una disutilità maggiore rispetto al tempo trascorso in auto.

I ricercatori hanno inoltre attuato un'ulteriore distinzione basandosi sulle classi di reddito, attraverso una differente valutazione monetaria del tempo per le diverse classi.

Un'altra considerazione importante è quella relativa al tempo trascorso in viaggio: è ragionevole ritenere che la disutilità generata dal tempo trascorso a bordo del proprio veicolo sia lineare; non è altrettanto ragionevole, però, accettare la stessa condizione per la modalità pedonale. Ha senso infatti ritenere che la disutilità relativa al tempo trascorso nella modalità pedonale cresca in modo più che proporzionale. Il soggetto che si sposta all'interno dell'area urbana, escludendo il caso del turista che intende ammirare la città, può essere disposto a camminare per 5 - 10 minuti percependo una certa disutilità, ma si suppone che oltre questo tempo, la disutilità cresca più che proporzionalmente. È anche ragionevole ritenere che vi sia una soglia di tempo oltre il quale l'utente non sia più disposto a spostarsi a piedi.

L'analisi del problema dell'individuazione dei coefficienti può essere un passaggio importante per confrontare i due approcci descritti in questo lavoro.

Tuttavia, non è possibile, o quantomeno non è del tutto significativo, fare un confronto diretto tra i due approcci, poiché i lavori in questione affrontano due problemi diversi tra loro nelle ipotesi.

Come già accennato, la scelta dei coefficienti gioca un ruolo fondamentale nell'approccio proposto da Modesti - Sciomachen, ed è di rilievo assai maggiore rispetto a quanto non sia nel lavoro di Opananon - Miller-Hooks. Osserviamo che, nell'analisi multimodale, un primo problema è quello di valutare il rapporto di preferenza che c'è tra le diverse modalità; è altresì importante osservare che questo rapporto non è costante né tra diversi individui, né per uno stesso individuo. Questo perché è intuibile che ci sia una certa differenza nell'utilità-disutilità data dal prendere un autobus pieno di gente nell'ora di punta o un autobus in un orario di calma, magari con diversi posti a sedere liberi; quando il traffico è bloccato, se in molte strade è presente la corsia preferenziale per i mezzi pubblici, se il numero di mezzi che servono una determinata linea è sufficiente a garantire una certa frequenza, per certe categorie di persone, la modalità pubblica può ritenersi preferibile a quella privata.



Nel loro lavoro, Modesti e Sciomachen fanno precisamente riferimento alla rete urbana del centro di Genova, tralasciando quindi alcuni aspetti e sottolineandone altri specifici del caso studiato, a discapito del problema generale, ma a vantaggio dell'analisi condotta.

Genova è una città con una conformazione particolare e con problemi del tutto singolari, per cui si rendono necessari, in un'analisi approfondita, specifici aggiustamenti nelle ipotesi e nei vincoli del problema.

Condizioni quali l'impossibilità di servire buona parte delle zone con il servizio pubblico in maniera efficiente ed efficace, l'enorme difficoltà, dovuta a carenze di spazio, di creare parcheggi di interscambio, la conformazione propria delle strade, talvolta strette, tortuose e ripide, insieme rendono, allo stato attuale delle cose, la modalità privata preferibile a quella pubblica, così come argomentato dai due ricercatori italiani.

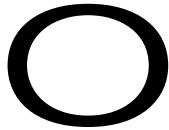
Al contrario, Opatanov e Miller-Hooks, nella loro analisi, hanno privilegiato l'aspetto generale del problema, proponendo un approccio molto generico e trascurando completamente il problema dell'assegnazione dei coefficienti necessari per ponderare le variabili decisionali.

In questo approccio, ad ogni variabile decisionale è assegnato un coefficiente numerico che rappresenta il peso della variabile sulla funzione di utilità; i coefficienti sono tali per cui la loro somma sia 1.

Il primo problema di questo tipo di approccio è dato dal fatto che le variabili esprimono risultati in differenti unità di misura, tra loro non confrontabili; in sostanza, i ricercatori non si sono posti il problema di omogeneizzare le unità di misura, rendendo scarsamente applicabile, allo stato attuale, algoritmi di questo tipo in reti di trasporto urbano.

Il lavoro offre un approccio teorico e computazionale, senza tenere conto effettivamente delle specificità delle reti di trasporto urbano. Quella che, a mio avviso, è la carenza più grave è proprio la totale assenza dell'analisi del problema dell'assegnazione dei coefficienti e dell'omogeneizzazione delle unità di misura. Un'ulteriore riprova dell'inconsistenza pratica del lavoro presentato è proprio data dalla sperimentazione svolta su convenienti reti generate aleatoriamente con il solo obiettivo di sperimentare l'efficienza computazionale degli algoritmi.

### 3.5 Una possibile integrazione



osservando quanto analizzato in questo lavoro, appare sensato, a questo punto dell'opera, provare ad amalgamare i due progetti discussi, al fine di ottenere un modello che possa godere dei vantaggi dati dall'utilizzo dei due metodi congiunti.

Consideriamo che abbiamo due possibilità di azione, strettamente legate l'una all'altra:

- ⊕ Considerare una funzione di disutilità per ogni variabile, omogeneizzarle per quanto riguarda l'unità di misura e applicare gli algoritmi proposti da Opananon - Miller-Hooks.
- ⊕ Considerare la funzione di utilità (o disutilità) come una funzione reale in più variabili ed applicare l'algoritmo di Dijkstra.

Poiché il problema della scelta dei coefficienti si ripresenta in maniera identica a quanto visto, questo tipo di approccio ha la sola finalità di isolare la questione, con l'obiettivo di focalizzare gli sforzi su quello che abbiamo osservato essere l'aspetto cruciale in questo tipo di analisi.

Svolgendo l'analisi attraverso il primo approccio, una possibile via di soluzione del problema può essere individuata dai seguenti passaggi:

1. Uniformare le unità di misura delle funzioni di disutilità riconducendole ad una unica.
2. Far scegliere all'utente, in base ai suoi criteri di preferenza, il valore dei

coefficienti  $w^k$ , pesi delle variabili, in modo che sia  $\sum_{k=1}^r w^k = 1$ .

Agendo in questo modo, sono possibili entrambi gli approcci, ma è importante tenere presente che l'analisi di una funzione aggregata come questa, non consente la possibilità di tenere in considerazione le interdipendenze tra variabili.

Ciò significa che, se le variabili non sono tra loro indipendenti, è possibile che questo metodo sia poco adeguato all'utilizzo in un approccio del secondo tipo.

L'importanza delle due singole fasi è interpretabile in maniera congiunta.

Per uniformare le unità di misura, il metodo che giudico più opportuno è il ricorso ad un'interpolazione delle preferenze individuali di un adeguato campione di popolazione; questo tipo di procedimento consente di attribuire un valore verosimile, presumibilmente accettato come tale da buona parte degli utenti, sul quale questi sarebbero in grado di misurare le proprie preferenze.

Ricondotte le unità di misura ad una unica, ogni utente può liberamente decidere quale peso attribuire, in base alle proprie preferenze personali, a ciascun fattore che influenza la sua funzione di disutilità.

Questo aspetto è particolarmente determinante per il tipo di ricerca affrontata, poiché il rilievo che può assumere nelle sue applicazioni è elevato.

La versatilità con cui possono essere applicati questi algoritmi è ampia; in termini di schemi di decisione, si ha la possibilità di adattare questi algoritmi secondo le diverse esigenze.

In termini pratici, è assai diverso il peso che assume la variabile tempo per il conducente di un mezzo di emergenza come un'autoambulanza o per il conducente di un camion a rimorchio, la cui prima preoccupazione è data dal trovare strade adeguate alle dimensioni del mezzo.

All'interno dei più recenti schemi decisionali, sta prendendo sempre più piede questo aspetto, cioè la possibilità di "personalizzare" i processi decisionali in base alle preferenze personali. Poiché, in merito soprattutto ai lavori analizzati in questa tesi, l'applicazione di procedure simili richiede una precisa stima dei parametri generali, ritengo che valga la pena approfondire questo tema nell'ambito della ricerca.

È altrettanto importante, ai fini dell'aspetto strettamente computazionale, limitare il numero di variabili a quelle essenziali dal punto di vista del processo decisionali, eliminando variabili di scarso peso; queste comporterebbero infatti un aumento della complessità computazionale più svantaggioso di quanto possa essere vantaggioso, in termini di verosimiglianza, il loro computo.

### 3.6 Conclusioni

**P**er terminare l'analisi fin qui condotta, è necessario, a mio parere, comprendere quali siano i punti salienti dei due lavori analizzati, determinarne i principali pregi ed evidenziarne le carenze, in modo tale da costituire il punto di partenza per lavori futuri che possano affrontare in maniera completa ed efficiente il problema del cammino minimo in reti MSTV.

Come già sottolineato, e come facilmente intuibile, il principale problema è quello discusso nel paragrafo precedente, ovvero la scelta dei coefficienti da attribuire alle variabili.

Sotto questo punto di vista, il lavoro di Modesti - Sciomachen, anche se in via puramente teorica, affronta il problema, ed è facile, leggendo il lavoro proposto, rendersi conto di come questo punto sia la chiave principale in questo tipo di problemi.

Dal punto di vista computazionale, il lavoro di Opananon - Miller-Hooks presenta il pregio di proporre due algoritmi efficienti in termini pratici, rendendo possibile l'applicazione di algoritmi esatti a problemi di dimensioni anche già realistiche; la comparabilità dei lavori qui diventa però scarsa, poiché vista la data delle due pubblicazioni, è facile arguire che i risultati sono viziati dallo stato delle tecnologie; dal momento in cui è stato svolto il lavoro di Modesti - Sciomachen allo sviluppo degli algoritmi ALEDS vi è stato uno sviluppo esplosivo nella velocità dei calcolatori elettronici, passando da processori 486 con una frequenza pari a pochi Megahertz a processori che superano i 3000, per non parlare della possibilità di utilizzare i processori in parallelo, aumentando smisuratamente la capacità computazionale.

Alla luce di queste ultime considerazioni, appare evidente come il problema dell'assegnazione dei coefficienti rappresenti il punto essenziale su cui la ricerca deve essere concentrata, potendo contare su algoritmi già efficienti e su calcolatori sempre più potenti.

# *Bibliografia*

- ⊗ Carl B. Boyer, *Storia della matematica*, 1990, Mondadori, Milano.
- ⊗ Ilaria Carpentieri, *Tesi di laurea*, 2004.
- ⊗ Domenico Lenzi, *Leonardo Eulero e i ponti di Königsberg*, *Lettera Matematica Pristem*, in corso di pubblicazione.
- ⊗ Sathaporn Opananon, Elise Miller-Hooks, *European journal of operational research*, in corso di pubblicazione.
- ⊗ Paola Modesti, Anna Sciomachen, *A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks*, *European Journal of operational research* 111, 1998, pag. 495 – 508.
- ⊗ Antonio Sassano, *Modelli e algoritmi della ricerca operativa*, 1999, FrancoAngeli, Milano.
- ⊗ [www.matematicamente.it](http://www.matematicamente.it)

*Non posso esimermi dal ringraziare dovutamente la professoressa Anna Sciomachen per l'attenzione dedicatami durante lo svolgimento di questo lavoro.*

*Dovuti ringraziamenti sono anche quelli che vanno alla professoressa Angela Berto, che con i suoi insegnamenti mi ha permesso di giungere fino a questo punto.*

*Doveroso ancora è il ringraziamento che va alla professoressa Giuseppina Fenaroli, perchè ora più di allora ho ricordato di saper camminare, e grazie a lei ho capito di essere in grado anche di correre.*

*Al professor Antonio Garibaldi che, con i suoi racconti accademici, mi ha condotto verso la ricerca delle radici della logica, va un grazie sentito.*

*A tutte le persone che hanno creduto in me in questi anni e mi hanno sostenuto, grazie per il loro prezioso appoggio.*

*A tutte le persone che mi hanno criticato, per avermi aiutato ad osservare i miei errori, grazie per l'aiuto.*

*A tutte le persone che in me non hanno mai creduto, che hanno sempre scommesso contro di me e per cui sono solo un cretino, il grazie più grande, poiché la volontà di sbeffeggiarli dopo esser giunto al traguardo mi ha dato più volte la forza per farcela.*

# Indice

Introduzione.....	I
Capitolo 1.....	III
1.1 – L'origine della teoria dei grafi.....	IV
1.2 – Introduzione alla teoria dei grafi.....	VII
1.3 – Il problema del cammino minimo.....	X
1.4 – Algoritmi euristici.....	XII
1.5 – Il problema del cammino minimo nelle reti urbane.....	XIV
1.6 – Introduzione al problema MSTV.....	XVIII
1.7 – Relazioni tra i cammini Pareto-ottimali a priori e adattivi.....	XXI
Capitolo 2.....	XXIV
2.1 – Notazione nelle reti e definizione del problema.....	XXV
2.2 – L'algoritmo APS.....	XXVII
2.3 – Considerazioni sull'algoritmo APS.....	XXXIII
2.4 – L'algoritmo ALEDS I.....	XXXV
2.5 – Considerazioni sull'algoritmo ALEDS I.....	XXXVIII
2.6 – L'algoritmo ALEDS II.....	XXXIX
2.7 – Considerazioni sull'algoritmo ALEDS II.....	XL
2.8 – Esperimenti su convenienti reti standard.....	XLII
2.9 – Conclusioni.....	XLV
Capitolo 3.....	XLVII
3.1 – La riduzione al problema a criterio singolo.....	XLVIII
3.2 – Il lavoro di Modesti – Sciomachen.....	XLIX
3.3 – La misura di utilità proposta.....	LI
3.4 – Il problema della scelta dei coefficienti.....	LIII
3.5 – Una possibile integrazione.....	LVI
3.5 – Conclusioni.....	LVIII
Bibliografia.....	LIX
Ringraziamenti.....	LX