



**UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA**

CORSO DI LAUREA IN INGEGNERIA GESTIONALE
DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

TESI DI LAUREA

**L'algoritmo "Tabu Search"
per l'ottimizzazione,
teoria ed applicazioni**

Relatore: Prof. GIORGIO ROMANIN JACUR

Laureando: ALESSANDRO SOLDANO

Matricola 579824-IG

ANNO ACCADEMICO 2011 – 2012

Indice

1	INTRODUZIONE	3
1.1	Cenni storici.....	4
1.2	Applicazioni di Tabu Search	5
1.3	Scrittura dell'algoritmo	6
2	CARATTERISTICHE PRINCIPALI.....	9
2.1	Memoria Adattabile	10
2.2	Esplorazione Attiva.....	11
3	MEMORIA A BREVE TERMINE	13
3.1	Gestione della memoria a breve termine	13
3.2	Strutture di memoria basate sulla recenza	17
3.3	Strategia della lista di candidati	20
3.4	Livelli di aspirazione	25
4	MEMORIA A LUNGO TERMINE.....	27
4.1	Strategie di Intensificazione.....	27
4.2	Strutture di memoria basate sulla frequenza	31
4.3	Strategie di diversificazione	33
5	OSCILLAZIONE STRATEGICA	39
5.1	Strutture di memoria dell'oscillazione strategica	42
6	PERCORSO DI RICOLLEGAMENTO	47
6.1	Considerazioni sul Percorso di ricollegamento	51
7	ESEMPIO APPLICATIVO: problema dell'albero con k-rami minimizzato	55
8	CONCLUSIONE	67
9	BIBLIOGRAFIA.....	69
	<i>Ringraziamenti</i>	71

1 INTRODUZIONE

Questa tesi analizza l'algoritmo Tabu Search (TS) sia da un punto di vista teorico che con un semplice esempio di ottimizzazione di un grafo. È una tecnica che guida una procedura di ricerca euristica locale per esplorare lo spazio delle soluzioni superando gli ottimi locali. Ad ogni iterazione si considera un intorno dove sono state rimosse alcune soluzioni, definite tabù, sulla base di una memoria del processo di ricerca effettuato fino a quella iterazione e le soluzioni tabù non sono quindi raggiungibili all'iterazione successiva. Per "tecnica euristica" si intende la ricerca di una buona soluzione, accettabile senza necessariamente raggiungere la soluzione migliore assoluta, con un risparmio di risorse (tempo) ed un risultato comunque soddisfacente. Questa tecnica è stata introdotta da Glover nel 1986.

Gli algoritmi euristici possono essere classificati in:

- algoritmi costruttivi: sfruttano le proprietà strutturali delle soluzioni ammissibili o i risultati della programmazione matematica
- algoritmi di ricerca locale: con uno schema di ricerca dello spazio delle soluzioni, partono da una soluzione euristica iniziale, si ricercano possibili miglioramenti nel vicinato (neighborhood), si esegue un'iterazione fino a quando non esistono ulteriori miglioramenti
- algoritmi metaeuristici: si estende la ricerca locale con l'adozione della diversificazione (ricerca in regioni diverse dello spazio delle soluzioni) e dell'intensificazione (ricerca delle migliori soluzioni nella regione corrente)

Gli algoritmi costruttivi sono in genere molto legati al problema specifico che vogliono risolvere mentre le tecniche di ricerca locale, e ancor più i metaeuristici, costituiscono un contesto generale, adatto ad essere utilizzato per la risoluzione di problemi di tipo anche diverso.

Sono state presentate in letteratura diverse tipologie di tecniche metaeuristiche, fra le quali:

- Simulated Annealing (Kirkpatrick, Gelatt e Vecchi 1983)
- Tabu Search (Glover 1986)
- Algoritmi Genetici (Rechenberg 1973, Holland 1975)
- Variable Neighborhood Search (Mladenovic, Labb e Hansen 1990)
- Ant Colony Optimization ACO (Coloni, Dorigo e Maniezzo 1992)
- Scatter Search (Glover 1965)

- Path Relinking (Glover 1965)
- Greedy Randomized Adaptive Search Procedure GRASP (Feo e Resende 1989)
- Guided Local Search GLS (Voudouris and Tsang 1990)
- Artificial Neural Networks (Hopfeld e Tank 1985, Mc Culloch and Pitts 1922)
- Memetic Algorithms (Moscato 1989)

1.1 Cenni storici

La parola *tabu* (o *taboo*) si trova nel Tongano, una lingua della Polinesia, dove era usata dagli aborigeni dell'isola di Tonga per indicare cose che non possono essere toccate in quanto sacre. Oggi nei dizionari la parola indica “una proibizione imposta da regole sociali come misura protettiva” oppure come qualcosa “proibito perchè costituisce un rischio”. Questi significati correnti della parola si accompagnano meglio al tema analizzato in questa tesi della ricerca con tabù.

La forma di Tabu Search più in uso oggi si deve a Fred Glover, anche se idee simili erano state abbozzate in contemporanea da P. Hansen, mentre il concetto fondamentale di usare proibizioni per incoraggiare una diversificazione della ricerca appare in vari lavori precedenti, ad esempio nell'algorithmo di Kernighan–Lin per il partizionamento dei grafi.

I numerosi esperimenti computazionali effettuati dimostrano che la Tabu Search è ormai una tecnica di ottimizzazione affermata, che può competere con quasi tutti i metodi noti e che, grazie alla sua flessibilità, può battere molte delle procedure classiche di ottimizzazione. Insieme al Simulated Annealing e agli algoritmi genetici, la Tabu Search è stata indicata dal Committee on the Next Decade of Operations Research come tecnica estremamente promettente per il trattamento futuro di applicazioni pratiche. Può essere formalmente dimostrata la convergenza asintotica verso la soluzione ottima della Tabu Search. Recentemente U. Faigle e W. Kern hanno proposto una versione probabilistica della Tabu Search che converge quasi sicuramente ad un ottimo globale. Tali risultati teorici, sebbene interessanti, sono piuttosto irrilevanti per le applicazioni pratiche. L'obiettivo della Tabu Search, come di altre tecniche euristiche, è di trovare rapidamente soluzioni buone per problemi che spesso non permettono di determinare soluzioni ottimali in tempi ragionevoli.

Il successo diffuso nelle applicazioni pratiche di ottimizzazione ha stimolato negli ultimi anni una rapida crescita della ricerca con tabù. Le procedure TS che sono descritte in questa tesi incorporano elementi base, ibridi con altri euristici e metodi algoritmici che sono riusciti a

trovare soluzioni migliori a problemi di programmazione, sequenziamento, allocazione delle risorse, pianificazione degli investimenti, telecomunicazioni e molti altri settori. Alcune delle diverse applicazioni di TS sono indicate in seguito.

Tabu Search si basa sulla premessa che la soluzione dei problemi, al fine di qualificarsi come intelligente, deve incorporare una memoria adattabile e reagire durante l'esplorazione. L'uso della memoria adattabile contrasta con strutture "senza memoria", come quelle ispirate a fenomeni della fisica e della biologia, con quelle a "memoria rigida" esemplificate dal "branch and bound" e con simili correlate all'intelligenza artificiale. L'enfasi sull'esplorazione attiva (e quindi finalizzata) in TS, sia in condizione deterministica che probabilistica, deriva dal presupposto che una cattiva scelta strategica può dare più informazioni rispetto a una buona scelta casuale.

1.2 Applicazioni di Tabu Search

- Programmazione (tempo di attraversamento delle celle di produzione, elaborazione di una pianificazione, pianificazione della forza lavoro, programmazione Aula, programmazione di macchine, programmazione di processi produttivi flow-shop o job shop, sequenziamento e dosaggio)
- Telecomunicazioni (instradamento delle chiamate, imballaggio della larghezza di banda, posizionamento dell'Hub della struttura, percorso di assegnazione, struttura della rete per i servizi, pianificazione dello sconto al cliente, la non immunità architettonica, reti sincrone ottiche)
- Progettazione (progettazione a computer, reti con tolleranza di errore, struttura della rete di trasporto, spazio di progettazione architettonica, schema di coerenza, struttura di reti a canone fisso, problemi di taglio irregolare, pianificazione della disposizione aziendale)
- Produzione, Magazzino e Investimenti (produzione flessibile, produzione Just-in-Time, parte di selezione della capacità MRP, pianificazione di un inventario multi-voce, sconto per volume di acquisto, mescolamento degli investimenti fissi)
- Localizzazione e allocazione (posizionamento ed assegnazione delle diverse materie prime, assegnazione quadratica, assegnazione semi-quadratica, assegnazione multilivello generalizzata)
- Instradamento (instradamento di veicoli, capacità dell'instradamento, tempo di finestra per l'instradamento, instradamento diversificato, instradamento a flotta)

mista, problema del commesso viaggiatore, viaggio dell'acquirente, programmazione del convoglio)

- Logica ed Intelligenza Artificiale (massima soddisfacibilità, logica probabilistica, raggruppamento, modello di riconoscimento o classificazione, integrità dei dati, formazione di reti neurali, struttura di reti neurali)
- Ottimizzazione di grafi (il grafo di partizionamento, partizionamento di "clique" o cricca, problemi di cricca massima, pianificazione di massimizzazione grafica, problemi di P-medio, grafica dei colori)
- Tecnologia (inversione sismica, distribuzione di energia elettrica, progettazione di ingegneria strutturale, volume minimo di ellissoidi, costruzione della stazione spaziale, posizionamento di celle di circuito, ricerca Off-Shore di idrocarburi)
- Ottimizzazione Generale Combinatoria (programmazione in zero-uno, ottimizzazione della carica fissa, programmazione non lineare non convessa, strutture tutto o niente, programmazione a due livelli)

Tabu Search si occupa di trovare modi nuovi e più efficaci di sfruttare i concetti sopra espressi con lo sviluppo dei principi associati che possono migliorare l'attuazione di una ricerca intelligente. Come ciò si verifica, emerge da un nuovo mix strategico delle idee di base, con una migliore attuazione pratica nel raggiungimento delle soluzioni. Questo rende TS una zona fertile per la ricerca e per studi sperimentali.

1.3 Scrittura dell'algoritmo

Il più conosciuto metodo di ricerca nel "neighbourhood", usato per trovare con approssimazione il valore minimo di una funzione f con valore reale in un insieme S , è il "metodo di discesa".

-Step 1. Scegli una soluzione di partenza i in S .

-Step 2. Trova il migliore j in $N(i)$ (ad es. così che $f(j) \leq f(k)$ per ogni k in $N(i)$).

-Step 3. Se $f(j) \geq f(i)$ allora stop. Altrimenti imposta $i=j$ e vai allo Step 2.

Questo metodo chiaramente può fermarsi ad un minimo locale di f ma non ad un minimo globale. In generale, $N(i)$ non è definito esplicitamente: bisogna cercare j esplorando diverse direzioni rispetto ad i (per esempio gli assi delle coordinate).

Un primo passo verso la definizione di Tabu Search sta nel ridefinire il classico *Descent method*.

-Step 1. Scegli una soluzione di partenza i in S .

- Step 2. Genera un sottoinsieme V^* di soluzioni in $N(i)$.
- Step 3. Trova un ottimo j in V^* (per es. così ch  $f(j) \leq f(k)$ per ogni k in V^*) ed imposta $i=j$.
- Step 4. Se $f(j) \geq f(i)$ allora stop. Altrimenti vai allo Step 2.

Nel precedente metodo avevo considerato $V^*=N(i)$, ma questo spesso risulta essere troppo costoso in termini di tempo ed un appropriata scelta di V^* pu  dare un miglioramento sostanziale.

Con alcune modifiche possiamo avvicinarci ulteriormente alla definizione dell'algorithmo TS (i^*   la migliore soluzione trovata e k   il contatore di iterazioni).

- Step 1. Scegli una soluzione di partenza i in S . Imposta $i^*=i$ e $k=0$.
- Step 2. Imposta $k=k+1$ e genera un sottoinsieme V^* di soluzioni in $N(i,k)$.
- Step 3. Scegli un ottimo j in V^* (rispetto ad f o ad una funzione modificata f^\sim) ed imposta $i=j$.
- Step 4. Se $f(i) < f(i^*)$ allora imposta $i^*=i$.
- Step 5. Se   incontrata una condizione di arresto, allora stop. Altrimenti vai allo Step 2.

Si osserva che in questa formulazione   incluso il classico metodo di discesa.

Definendo: i come soluzione, m come numero delle mosse, $tr(i,m) \in Tr$ ($r=1,\dots,t$) come insieme di condizioni tab , $ar(i,m) \in Ar(i,m)$ ($r=1,\dots,a$) come condizioni di aspirazione (rilassamento dalla condizione di tab ) ed $f^\sim = f +$ Intensificazione + Diversificazione, posso infine scrivere:

Tabu search

- Step 1. Scegli una soluzione di partenza i in S . Imposta $i^*=i$ e $k=0$.
- Step 2. Imposta $k=k+1$ e genera un sottoinsieme V^* di soluzioni in $N(i,k)$ cos  ch  nessuna delle condizioni tab  $tr(i,m) \in Tr$ con ($r=1,\dots,t$) sia violata a meno di essere evasa con le condizioni di aspirazione $ar(i,m) \in Ar(i,m)$ con ($r=1,\dots,a$).
- Step 3. Scegli un ottimo $j=i+m$ in V^* (rispetto ad f od alla funzione f^\sim) ed imposta $i=j$.
- Step 4. Se $f(i) < f(i^*)$ allora imposta $i^*=i$.
- Step 5. Aggiorna le condizioni di tab  e di aspirazione.
- Step 6. Se si incontra una condizione di arresto, allora stop. Altrimenti vai allo Step 2.

2 CARATTERISTICHE PRINCIPALI

La base di TS può essere descritta come segue. Data una funzione $f(x)$ da ottimizzare su un insieme X , TS inizia allo stesso modo ordinario della ricerca locale, procedendo iterativamente da un punto (Soluzione) ad un altro fino a quando è soddisfatto un criterio di terminazione scelto. Ogni $x \in X$ ha associato un intorno di $N(x) \subset X$, e ogni soluzione x di $N(x)$ è raggiungibile da x con un'operazione chiamata "mossa". TS va oltre la ricerca locale utilizzando una strategia di modifica di $N(x)$ progredendo nella ricerca in modo efficace con la sua sostituzione con un altro intorno $N^*(x)$. Come precedente detto, un aspetto fondamentale del TS è l'uso di particolari strutture di memoria che servono per determinare $N^*(x)$ e quindi di organizzare il modo in cui viene esplorato lo spazio.

Le soluzioni ammesse a $N^*(x)$ da queste strutture di memoria sono determinate in diversi modi. Uno di questi, che dà a tabu search il suo nome, identifica le soluzioni incontrate nel corso di un determinato percorso (e implicitamente, relative soluzioni aggiuntive), e vieta loro di appartenere a $N^*(x)$ per la loro classificazione come "tabù". Il termine tabù supera il suo stretto significato etimologico in quanto questa restrizione potrà essere evasa quando saranno soddisfatte particolari condizioni di garanzia.

Il processo per cui le soluzioni acquisiscono uno status tabù ha diverse sfaccettature, destinate a promuovere un esame attento ed esaustivo di nuovi punti. Un modo utile per la visualizzazione e l'attuazione del presente processo è quello di immaginare di sostituire le iniziali classificazioni tabù delle soluzioni, cioè quelle escluse dalla scelta di $N^*(x)$, secondo la loro dipendenza dagli elementi che compongono lo stato tabù che introduce sanzioni per scoraggiarne in maniera significativa la scelta. Inoltre, le valutazioni tabù includono periodicamente anche incentivi per incoraggiare la scelta di altri tipi di soluzioni, a causa di livelli di aspirazione ed influenze a lungo termine. Va sottolineato che il concetto di un intorno di TS è diverso da quello utilizzato nella ricerca locale, abbracciando i tipi di mosse impiegate nei processi costruttivi e distruttivi (dove le basi per tali spostamenti sono quindi chiamati intorni costruttivi ed intorni distruttivi). Tali usi allargati del concetto di intorno rafforzano una prospettiva fondamentale di TS, che è quella di definire gli intorni in modo dinamico, che può includere o tenere conto simultaneamente di diversi tipi di mosse da meccanismi successivamente identificati.

Inizio illustrando in modo generale come TS si avvalga di memoria (e quindi di apprendimento di processi) per modificare le strutture degli intorni con cui lavora e per guidare il suo percorso

attraverso queste strutture. Con questa base, posso poi dare una visione più dettagliata dei componenti primari dell'algoritmo TS.

2.1 Memoria Adattabile

Schematizzo e riassumo le caratteristiche della Tabu Search che enfatizzano la memoria adattabile.

- Memorizzazione (lista vera e propria, vettore o matrice che memorizza l'iterazione della mossa).
- Memoria esplicita (non si ritorna sui propri passi, tempo computazionale elevato).
- Memoria attributiva (possibilità di ritornare sui propri passi, tempo computazionale limitato).
- Selettività (compresa la dimenticanza strategica).
- Astrazione e decomposizione (attraverso la memoria esplicita ed attributiva).
- Temporalità (cronologia di eventi, frequenza di eventi, differenziazione tra breve e lungo termine)
- Qualità e impatto (interesse relativo delle scelte alternative, entità dei cambiamenti di struttura o di relazioni vincolari).
- Contesto (interdipendenza spaziale, interdipendenza strutturale, interdipendenza sequenziale).
- Tabu Tenure (per quante iterazioni una mossa è considerata tabù, se corta l'intensificazione è preferita alla diversificazione, se è lunga è preferita la diversificazione, un compromesso si ha con la lunghezza dinamica) -Memoria a breve o lungo termine (la memoria a breve consiste essenzialmente nella Tabu List, la memoria a lungo termine è opzionale e tiene conto di informazioni su tutta la ricerca compiuta).

Mi soffermo più in dettaglio delle definizioni di memoria esplicita od attributiva. La memoria esplicita registra soluzioni complete, costituite in genere da soluzioni scelte visitate durante la ricerca (o vicini dell'intorno molto interessanti ma inesplorati in tali soluzioni). Queste soluzioni speciali sono introdotte ad intervalli strategici per ingrandire $N^*(x)$ e quindi fornire utili opzioni non in $N(x)$. La memoria attributiva, al contrario, registra le informazioni sugli attributi della soluzione che cambiano nel passaggio da una soluzione all'altra. Ad esempio, in un grafo o una rete, gli attributi possono consistere in nodi o rami che vengono aggiunti, eliminati o riposizionati con le mosse eseguite. In più con formulazioni astratte del problema, gli attributi possono corrispondere a valori delle variabili o funzioni. Utilizzata correttamente la memoria

attributiva consente di esercitare una serie di sottili influenze. A volte gli attributi sono anche strategicamente combinati in TS per creare altri attributi da utilizzare in tale memoria, come in procedure di associazione (hashing) o dalla suddivisione in blocchi collegati con l'AI (artificial intelligence) o i metodi di "ampliamento del vocabolario".

Poiché Tabu Search ha diversi componenti critici e l'integrazione può sembrare difficoltosa, alcune implementazioni sono basate solo su semplici idee tipicamente sviluppate in un'esposizione generale. Tuttavia, va sottolineato che la critica vale solo per un ristretto numero di componenti (ciascuno con alcune varianti chiave), e una volta assimilati, creare un quadro interconnesso è molto più efficace che concentrarsi solo su uno o due degli elementi separatamente.

2.2 Esplorazione Attiva

Elenco ora gli elementi che caratterizzano l'esplorazione attiva e che saranno successivamente sviluppati nella tesi.

- Restrizioni strategicamente imposte ed incentivi (condizioni di tabù e livelli di aspirazione, se una soluzione è tabù ma migliore dell'incombente la accetto lo stesso).
- Attenzione concentrata sulle regioni buone e sulle caratteristiche di una buona soluzione (processi di intensificazione).
- Caratterizzazione ed esplorazione di nuove regioni promettenti (processi di diversificazione, possibilità di usare penalità per attributi troppo ricorrenti, possibilità di applicare multi-start per regioni poco visitate, si possono usare matrici simili a quelle usate per la tabu list).
- Modelli di ricerca irregolari (oscillazioni strategiche).
- Integrazioni ed estensioni delle soluzioni (percorso di ricollegamento).

3 MEMORIA A BREVE TERMINE

Una distinzione importante nel TS nasce tra memoria a breve termine e memoria a lungo termine. Ogni tipo di memoria è accompagnata da proprie strategie particolari. La più comunemente utilizzata è la memoria a breve termine che tiene traccia degli attributi della soluzione che sono cambiati nel corso del recente passato, ed è chiamata memoria “Recency Based”: basata sulla recenza. La memoria RB è sfruttata assegnando un “tabù-attivo” per la designazione di attributi selezionati che si verificano in soluzioni già visitate di recente. Soluzioni che contengono elementi tabù-attivi, o particolari combinazioni di questi attributi, sono quelle che diventano tabù. Questo impedisce che vengano rivisitate certe soluzioni appartenenti a $N^*(x)$ di recente esplorazione. Allo stesso modo è impedito rivisitare altre soluzioni che condividono tali attributi tabù-attivi. L'uso di valutazioni tabù, con sanzioni assegnate ad insiemi di attributi del caso tabù-attivo, ha l'effetto di variare lo stato tabù per gradi.

3.1 Gestione della memoria a breve termine

Il processo è gestito mediante la creazione di una o più *tabu list*, che registrano gli attributi tabù-attivi e implicitamente o esplicitamente identificano il loro stato attuale. La durata di un attributo nel rimanere tabù-attivo (misurata in numero di iterazioni) si chiama *tabu tenure*, “livello di tabù” .I livelli di tabù possono variare per i diversi tipi o combinazioni di attributi e possono anche variare nei diversi intervalli di tempo o fasi di ricerca. Questo carico variabile consente di creare diversi tipi di compromessi tra strategie a breve e a lungo termine. Fornisce inoltre una forma dinamica e solida di ricerca. Un esempio di come questa memoria opera è rappresentato nella Figura 1 (pagina seguente). Il problema in questa illustrazione è di trovare un albero (ovvero un sottografo senza cicli) ottimizzato affinché, ad esempio, la somma delle lunghezze dei suoi rami sia minima, sul grafo con i nodi numerati da 1 a 7, come mostrato in figura.

Esempio: problema con obiettivo di ottimizzazione

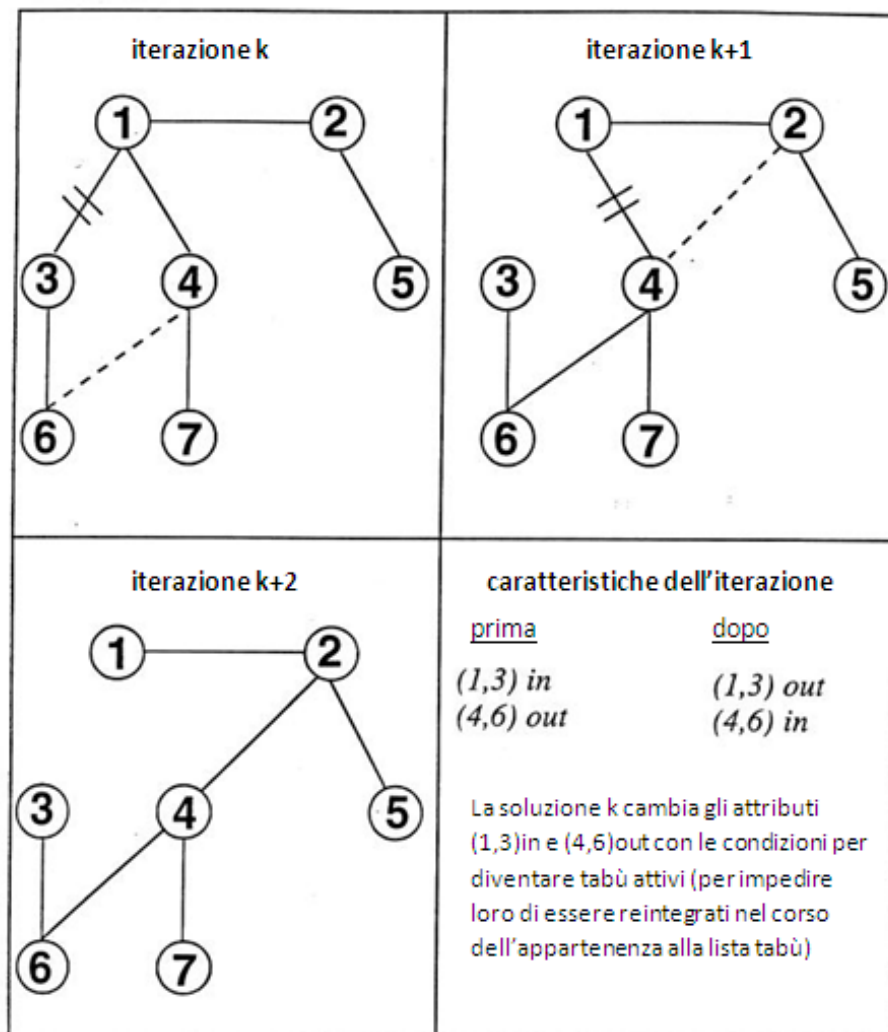


Figura 1

Per il caso in cui la funzione obiettivo è lineare il problema è molto semplice e si può poi supporre un'applicazione più complessa con obiettivo non lineare, come nella distribuzione di energia elettrica, di telecomunicazioni o in problemi di progettazione della rete.

Tutti rami possibili che uniscono le coppie di nodi saranno assunti disponibili per comporre l'albero e i tre sottografi illustrati per le iterazioni k, k+1 e k+2 (dove k è arbitraria) identificano gli alberi particolari generati nelle varie fasi di risoluzione del problema. Supponiamo che le mosse utilizzate per cambiare un albero in un altro (da qui che definiscono l'intorno $N(x)$, dove la soluzione x corrisponde ad un albero particolare) consistono nella selezione di un ramo da eliminare ed un altro da aggiungere, in modo che il risultato rimanga un albero. Il ramo tagliato deve trovarsi sul ciclo unico prodotto dall'introduzione del ramo aggiunto o, equivalentemente, il ramo aggiunto deve unire i due sottoalberi separati creati eliminando il ramo caduto. La mossa applicata all'iterazione k per produrre l'albero di iterazione k+1

consiste nel far cadere il ramo (1,3), e aggiungendo il ramo (4,6), come indicato, rispettivamente, dal ramo segnato con due linee incrociate e il ramo che è tratteggiato. La presenza del ramo (1,3) e l'assenza del ramo (4,6) nell'albero della iterazione k possono essere considerati come due differenti attributi della soluzione, che indichiamo con (1,3)_{in} e (4,6)_{out} come indicato nell'ultimo riquadro della Figura 1. Dato che questi sono gli attributi che cambiano a seguito della mossa, si qualificano per essere designati tabù-attivi e per essere utilizzati per definire lo stato di tabù delle mosse per future iterazioni. Supponiamo per il momento di classificare una mossa come tabù se uno qualsiasi dei suoi attributi è tabù-attivo. Ad esempio, possiamo specificare che (1,3)_{in} dovrebbe essere tabù-attivo per 3 iterazioni, cercando di evitare che il ramo (1,3) sia nuovamente aggiunto alla struttura attuale per questa durata, e (4,6)_{out} dovrebbe essere tabù-attivo per 1 iterazione, cercando di evitare al ramo (4,6) di essere rimosso dall'albero corrente per questa durata. Queste condizioni cercano effettivamente di evitare un percorso inverso rispetto ai particolari cambiamenti dati dalla mossa. I livelli tabù indicati di 3 e 1, naturalmente, sono molto piccoli e sarà poi discusso come tali livelli tabù possano essere scelti in modo appropriato. Tuttavia, le motivazioni per cui dare un valore più grande a (1,3)_{in} che a (4,6)_{out} è importante. In particolare, nel nostro esempio, esistono molti rami che possono essere aggiunti all'albero come parti di una mossa per creare una nuova struttura, ma esistono un po' meno rami che possono essere abbandonati in quanto parte di una tale mossa, dal momento che tutti i rami non appartenenti all'albero sono disponibili da aggiungere ma solo i rami dell'albero sono disponibili per essere tagliati. Così, rendendo (1,3)_{in} tabù-attivo, che impedisce al ramo (1,3) di essere aggiunto, è molto meno restrittivo che rendere (4,6)_{out} tabù-attivo, che impedisce al ramo (4,6) di essere tagliato. Detto in altro modo, impedendo ad un ramo di essere aggiunto si esclude un minor numero di mosse rispetto all'impedire ad un ramo di essere tagliato. In generale, quindi, il livello di tabù di un attributo deve dipendere dalla restrittività della condizione tabù associata.

La terminologia utilizzata in questo esempio può essere usata per fare riferimento semplicemente ai rami (1,3) e (4,6) come attributi del movimento, poiché la condizione di essere dentro o fuori è sempre automaticamente conosciuta dalla soluzione corrente. Quindi, possiamo semplicemente dire che questi due rami sono tabù-attivi, con livelli diversi. Se si considera una mossa che aggiunge il ramo (1,3) per l'albero, è necessario solo per verificare se questo ramo è tabù-attivo, senza tenere memorie separate a seconda se il ramo è presente o assente dalla struttura ad albero.

Attributi più complessi possono essere utilizzati per determinare lo stato tabù delle mosse, ma è necessaria attenzione. Tali attributi devono essere trattati come proprietà delle soluzioni piuttosto che proprietà delle mosse, poichè è da evitare il ciclismo durante il periodo che gli

attributi sono tabù-attivi. Il ciclismo è normalmente impedito per durate molto più lunghe ed in genere è eliminato del tutto per scelte ragionevoli dei livelli tabù. Ma evitare il ciclismo non è l'unico scopo della memoria basata sulla recenza.

Illustro di nuovo con riferimento alla Figura 1: supponiamo che l'iterazione k abbia creato una limitazione tabù secondo la quale una mossa che unisce i due attributi di aggiunta di (1,3) ed eliminazione di (4,6) sia tabù, cioè evitare l'inversione del movimento applicato a iterazione k . Poi, all'iterazione $k+1$ con questa condizione tabù-attiva, potremmo scegliere la mossa di aggiunta del ramo (2,4) e la caduta del ramo (1,4), come mostrato nella Figura 1, per produrre l'albero di iterazione $k+2$. Allo stesso modo, si stabilisce che la mossa di retromarcia, che combina gli attributi di aggiunta (1,4) e la caduta (2,4), sia tabù. Ora, a partire dall'albero di iterazione $k+2$, siamo in grado di eseguire due mosse aggiuntive in successione, la prima composta da l'aggiunta di (1,3) e la caduta (2,4), e la seconda composta dall'aggiunta di (1,4) e la caduta di (4,6). Anche se nessuna di queste mosse viola le restrizioni tabù, il risultato è quello di produrre l'albero originale all'iterazione k .

E' quindi importante sottolineare la distinzione tra rendere tabù-attivi gli attributi delle mosse (come nell'esempio immediatamente precedente) e rendere tabù-attivi gli attributi delle soluzioni. Così, ad esempio, potremmo avere posto invece che all'iterazione k l'attributo composto di (1,3)in e (4,6)out renderà una mossa tabù se la mossa crea una soluzione con questo attributo. Ciò eviterà il fenomeno del ciclismo appena indicato, ma il controllo di attributi composti all'interno di soluzioni di solito richiede più memoria che lo sforzo di controllo per attributi semplici.

Un'alternativa più semplice e più efficace è quella di esprimere le restrizioni tabù in termini di congiunzioni di attributi semplici. Così, per esempio, si potrebbe prevedere per una mossa di essere tabù solo se tutti, o anche solo alcuni, dei suoi attributi che compongono la soluzione sono tabù-attivi. Questo non richiede memoria aggiuntiva al di là di mantenere una memoria che rivela lo stato di tabù-attività dei singoli attributi, ed anche impedisce il ciclismo per tutta la durata dello stato attraversato dal tabù-attivo.

Tale approccio produce condizioni tabù meno restrittive di quelle a base di disgiunzioni, dove una mossa è tabù se uno dei suoi attributi è tabù-attivo, e quindi fornisce una maggiore flessibilità per la scelta di mosse. Nelle situazioni in cui tale flessibilità può essere opportunamente sfruttata, in particolare nelle fasi di intensificazione in certe aree di ricerca, come descritto più avanti, può essere utile identificare soluzioni secondarie di attributi per inserire restrizioni congiuntive. Ad esempio, nella figura è presente un altro tipo di attributo che è facilmente accessibile ed è aggiornato: il grado del nodo dei terminali di rami aggiunti ed eliminati, cioè il numero di spigoli degli alberi che si incontrano in questi punti di fine dei nodi.

La creazione di restrizioni tabù che incorporano congiunzioni in base ad attributi fornisce opzioni aggiuntive senza un significativo aumento nella trasformazione e con un aumento generalmente tollerabile in memoria.

Come base per identificare gli attributi che possono essere utilizzati nei modi che ho illustrato, una possibilità è quella di considerare l'uso di attributi creati, che risultano dalle funzioni di selezione di altri elementi semplici e identificano i valori di queste funzioni come attributi per definire le restrizioni tabù. La funzione obiettivo si qualifica come una tale funzione, ma meritano considerazione anche altre possibilità. La quantità di memoria aggiuntiva dipende dal numero di valori pertinenti delle funzioni che possono essere comprese in intervalli. In alternativa, questa memoria può essere fatta dipendere dalla lunghezza dei possessi tabù di tali valori (se viene utilizzata una lista circolare per registrare i valori generati su questi possessi), anche se questo può comportare un maggiore impegno per verificare le condizioni tabù. Tali preoccupazioni sono al cuore delle proposte di associazione e suddivisione in blocchi. Sottolineo ancora una volta, tuttavia, che l'uso di restrizioni sempre più allentate di tabù non è sempre auspicabile, poiché le restrizioni più forti hanno l'effetto di creare un certo vigore nel processo di ricerca, evitando soluzioni simili e soluzioni ripetute. Tuttavia la capacità di attingere a restrizioni tabù meno rigorose può offrire vantaggi nelle fasi dedite alla ricerca più a fondo nelle regioni altamente fertili.

Ho descritto in modo approfondito questi aspetti della creazione di restrizioni tabù che dipendono dall'attribuzione della memoria, perché gli attributi come quelli illustrati sono anche una base per i tipi di memoria TS diverse dalla memoria basata sulla recenza. Al tempo stesso, però, è lasciata aperta la questione di progettazione di strutture di memoria specifiche per gestire convenientemente le restrizioni tabù. Esempi di tali strutture per la memoria basata sulla recenza, e relative norme di attuazione, sono riportate qui sotto.

3.2 Strutture di memoria basate sulla recenza

Indico ora alcune strutture di memoria basate sulla recenza comunemente usate per identificare gli attributi che sono tabù-attivi e per la determinazione degli stati tabù di soluzioni contenenti tali attributi. Sia $S = \{1, 2, \dots, s\}$ ad indicare un insieme di indici per un insieme di attributi di una soluzione. Ad esempio, gli indici $i \in S$ possono corrispondere ad indici di x_i variabili in zero-uno, o possono essere indici di rami che possono essere aggiunti o eliminati da un grafo. Più precisamente, gli attributi di riferimento di S in questi due casi consistono in valori specifici assegnati alle variabili o gli specifici stati aggiunti/eliminati adottati dai rami. In

generale, un indice $i \in S$ può riassumere più informazioni dettagliate: ad esempio, facendo riferimento ad una coppia ordinata (j, k) che riassume l'assegnazione di un valore $x_j = k$. Quindi, l'indice i può essere visto come un vantaggio di notazione per la rappresentazione di una coppia o un vettore, ecc..

Per l'esempio specifico, si supponga che ogni $i \in S$ corrisponda ad una x_i variabile in zero-uno. Non mi dilungherò a scrivere $(i, 0)$ e $(i, 1)$ ad identificare i due attributi associati $x_i=0$ e $x_i=1$ in quanto conoscendo il valore corrente di x_i sappiamo anche il suo unico valore alternativo. Per registrare la base attuale di informazioni di TS per ogni variabile, teniamo traccia delle iterazioni con un contatore di iterazioni denotato *current_iteration*, che inizia da 0 e aumenta di 1 ogni volta che viene effettuata una mossa. Quando si esegue una mossa che provoca il cambiamento del valore di una variabile x_i , si registra *tabu_start(i)=current_iteration* subito dopo l'aggiornamento del contatore di iterazione. Ciò significa che se la mossa è portata a $x_i=1$, allora l'attributo $x_i=0$ diventa tabu-attivo all'iterazione *tabu_start(i)*. Inoltre, si prevede che questo attributo rimanga tabu-attivo per un certo numero di iterazioni pari al *tabu_tenure(i)*, il cui valore sarà determinato in un modo che sarà presto indicato. Così, in particolare, il criterio basato sulla recenza tabù dice che il valore precedente di x_i è tabu-attivo in tutte le iterazioni in modo che $tabu_start(i) + tabu_tenure(i) = current_iteration$.

Una volta aumentata *current_iteration* al punto in cui questa disuguaglianza non regge più, x_i non sarà più tabu-attivo al suo valore precedente e quindi non può essere scoraggiato il ricevimento di tale nuovo valore. Il valore *tabu_start(i)* può essere impostato a 0 prima di iniziare la procedura, come una convenzione per indicare l'assenza di memoria precedente. Poi abbiamo automaticamente evitato di assegnare uno stato tabu-attivo ad ogni variabile con *tabu_start(i)=0* in quanto il valore di partenza per la variabile x_i non è ancora stato cambiato. Per comodità di seguito farò riferimento ad una variabile come x_i tabu-attivo fermo restando che la condizione di tabu-attivo si applica ad un attributo specifico associato all'attributo $x_i=K$ dove k è l'ultimo valore assegnato in precedenza a x_i . Se solo una variabile cambia il suo valore su un'iterazione, una mossa può essere classificata tabù ogni volta che cambia il valore di una variabile tabù-attiva. Tuttavia, se due variabili cambiano i loro valori, dove si è impostato su 0 e l'altra a 1, poi ci sono diverse scelte. Ad esempio, la mossa può essere designata tabù se:

- (A) entrambe le variabili sono tabù-attive;
- (B) almeno una variabile è tabu-attiva;
- (C) la variabile che cambia da 0 a 1 è tabu-attiva.

La possibilità che lo stato tabù dovrebbe dipendere da un particolare cambiamento di valori, come in (c), può anche essere riflessa dando a $tabu_tenure(i)$ un valore diverso a seconda del valore assegnato a x_i .

La scelta di un valore preferito per $tabu_tenure(i)$ è abitualmente sulla base di test empirici, a partire dal considerare un valore comune per tutti gli attributi (o per tutti gli attributi in una classe specifica). L'esperienza dimostra che le opzioni possono essere rapidamente ridotte ad una gamma in cui ogni valore dà buoni risultati, soprattutto se il valore viene considerato come il centro di un piccolo intervallo in cui $tabu_tenure(i)$ è vario, sistematicamente o casualmente.

I valori di livello tabù per classi date di problemi in genere possono essere espressi come una semplice funzione del numero totale di attributi (come frazione o multiplo della radice quadrata di s). Per una maggiore raffinatezza, allora tali valori possono essere differenziati a seconda del tipo di attributi come per esempio, in base alle assegnazioni $x_i = 0$ o 1 , e secondo le specifiche tipologie di variabili. Questi tipi di perfezionamenti possono essere fatti adattandoli all'interno del processo di soluzione stessa, monitorando le conseguenze di scelte alternative. Ad esempio, Laguna et al. (1992) controllano la qualità dei movimenti associati a cambiamenti di un particolare attributo e variano il livello tabù degli attributi che partecipano a mosse dall'attrattiva maggiore o minore. In un altro tipo di approccio, Kelly et al. (1991) tengono traccia dei modelli di dei cambiamenti di valore della funzione obiettivo e modificano lo stato tabù di mosse quando il modello suggerisce la possibilità del ciclismo. Battiti e Tecchioli (1992) forniscono un metodo efficace che utilizza un'associazione che funziona come un indicatore di ciclismo, e modifica direttamente un valore complessivo del livello tabù come il processo di ricerca continua fino a quando questo valore è appena sufficiente per eliminare le tracce di ciclismo. Questo tipo di approccio può essere esteso, approfittando delle idee di suddivisione in blocchi di Woodruff (1993).

Una strategia dinamica con una base un po' diversa determina lo stato tabù senza fare affidamento ad un'assegnazione a tutti i tabù, ma tenendo conto di relazioni logiche nella sequenza di cambiamenti di attributi. Un adeguato riferimento a questi rapporti permette di stabilire in anticipo se un particolare cambiamento di corrente sia in grado di produrre il ciclismo e quindi generare restrizioni tabù che sono necessarie e sufficienti per evitare di tornare alle soluzioni precedenti (Glover 1990). Un'assegnazione tabù piccola introduce vigore extra nella ricerca, dal momento che la prevenzione del ciclismo non è l'unico obiettivo della memoria basata sulla recenza. In aggiunta, una "durata di memoria limitata" riduce i costi e fornisce maggiore flessibilità dove talvolta può essere preferibile rivisitare soluzioni precedentemente incontrate. Ciò significa sfruttare le interdipendenze logiche e fornire anche

informazioni utili per le strategie di diversificazione innovative. Le implementazioni sono state sviluppate da Dammeyer (1991) e Voss (1992).

Mentre esistono interessanti opportunità per l'applicazione di forme avanzate di memoria basata sulla recenza di TS, è da notare che le forme più semplici spesso funzionano abbastanza bene. Questo motiva l'uso di semplici tipi di memoria come base per lo sviluppo iniziale di implementazioni di TS. L'esperienza con tali implementazioni può quindi suggerire la base per elaborazioni produttive. Questa caratteristica di TS, che permette di introdurre miglioramenti per tappe graduali, è particolarmente utile per procedere agli schemi che incorporano la memoria a lungo termine che illustrerò in seguito.

3.3 Strategia della lista di candidati

L'aspetto aggressivo di TS è rafforzato dalla ricerca della migliore mossa disponibile che può essere determinata con una quantità di sforzo adeguata. Dovrebbe essere tenuto presente che il significato di “migliore” non si limita alla valutazione della funzione obiettivo. Come già sottolineato, le valutazioni tabù sono colpite da sanzioni ed incentivi determinati dalla cronologia di ricerca. Essi sono anche influenzati da considerazioni di influenza come caratterizzato successivamente. Per le situazioni dove $N^*(x)$ è di grandi dimensioni o dei suoi elementi sono “costosi”, la strategia della lista dei candidati viene utilizzata per limitare il numero delle soluzioni esaminate con le date iterazioni.

Data l'importanza, TS attribuisce alla selezione di elementi di giudizio, le regole efficienti per generare e valutare buoni candidati che sono critici per il processo di ricerca. Strutture della memoria che danno aggiornamenti efficienti delle valutazioni della mossa da un'iterazione all'altra e per ridurre lo sforzo di trovare le mosse migliori o nelle vicinanze, sono spesso parte integrante delle implementazioni di TS, anche dove strategie di liste dei candidati non sono usate in modo esplicito. L'aggiornamento intelligente può sensibilmente ridurre i tempi di soluzione e l'inclusione di esplicite strategie di candidati in problemi di grandi dimensioni può significativamente aumentare i conseguenti benefici.

Ora considero insieme questi elementi a breve termine ed illustro come interagiscono nella Figura 2 (pagina seguente).

VALUTAZIONI SU TABU SEARCH (memoria a breve termine)

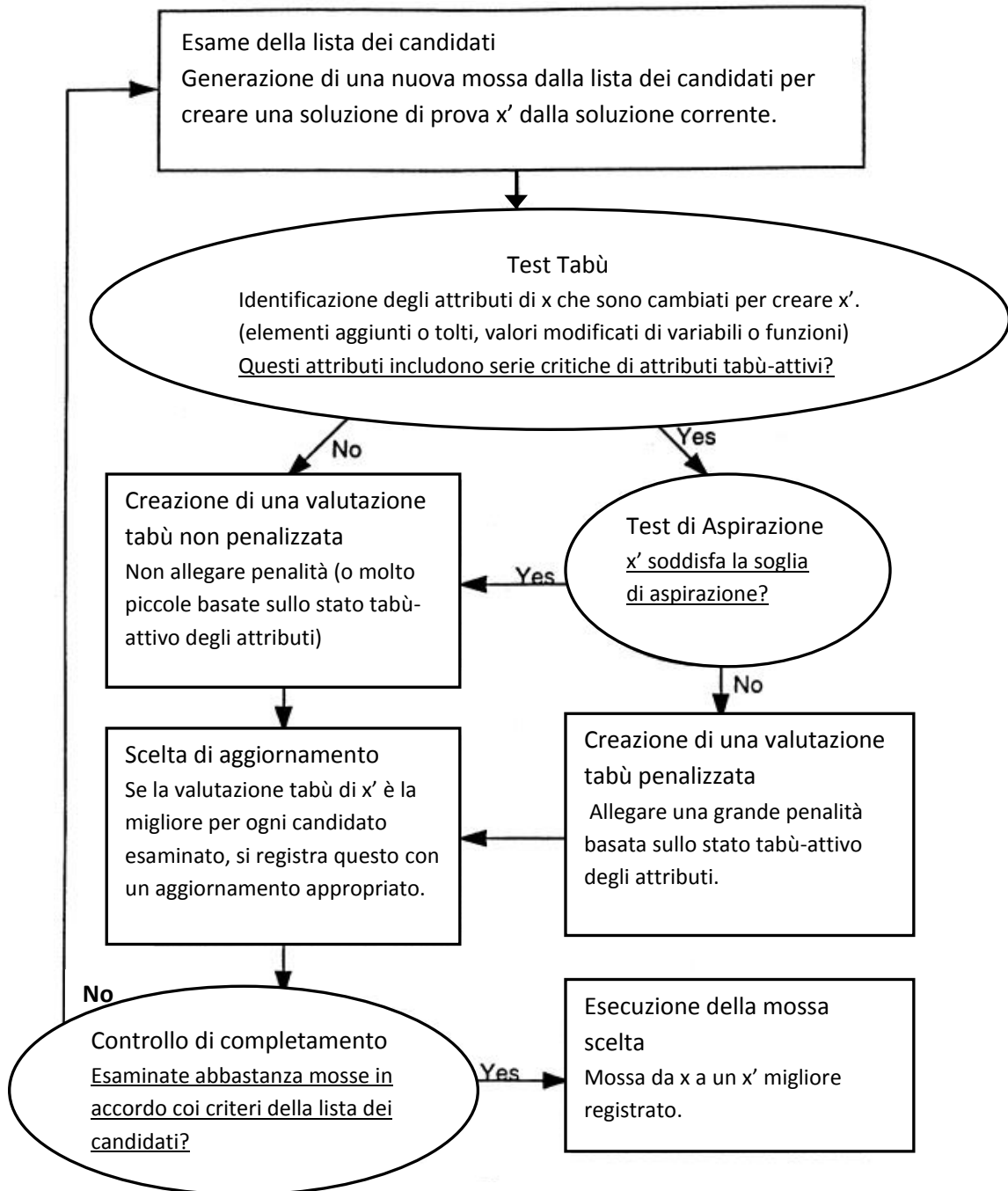


Figura 2

La rappresentazione delle sanzioni nella Figura 2 sia come "grandi" o "molto piccole" esprime un effetto di soglia. Nell'illustrazione della Figura 1, ho trattato lo stato tabù come un tipo di condizione tutto-o-niente, ma la differenziazione è chiaramente possibile, con riferimento ad un diverso numero di elementi tabù attivi o ad uno dei diversi livelli di considerazione tabù non scaduti. Lo stato tabù corrisponde generalmente ad utilizzare le sanzioni che forniscono una

valutazione fortemente deteriorata, oppure che è servita soprattutto a rompere i legami tra le soluzioni con le valutazioni più alte. La rottura del legame avviene per le mosse che non sono più tabù in un senso tutto-o-niente, consentendo un'influenza persistente, diminuita a seconda della durata dei livelli, che potrebbe altrimenti essere considerata scaduta.

Le sanzioni risultano nella scelta di una soluzione "meno tabù" e si può osservare che la sequenza del tabù test ed il test di aspirazione nella Figura 2 possono essere scambiati (cioè con l'impiego del test tabù solo se la soglia di aspirazione non è soddisfatta). Inoltre, la valutazione tabù può essere modificata con la creazione di incentivi in base al livello di aspirazione, così come è modificato con la creazione di sanzioni in base allo stato tabù. In questo senso, le condizioni di aspirazione e le condizioni di tabù possono essere concepite approssimativamente come "immagine speculare" della vicenda. La variante TS chiamata "tabu search probabilistico" segue un disegno corrispondente, con un componente di breve termine che può essere rappresentato dallo stesso schema. L'approccio tiene inoltre traccia delle valutazioni tabù generate durante il processo che determina la selezione di un movimento. Sulla base di questo sistema, la mossa è scelta probabilisticamente dall'insieme di quelle valutate o da un sottoinsieme dei migliori membri di questo insieme, la ponderazione si muove in modo che quelli con valutazioni più alte siano particolarmente favorite.

Sia la velocità sia la qualità delle soluzioni possono essere influenzate in modo significativo con l'uso di appropriate strategie di liste di candidati. Forse sorprendentemente, l'importanza di tali approcci è spesso trascurata, anche se sono fondamentali per l'importanza di TS nell'ambito di scelte intelligenti. Espongo ora degli esempi di alcune strategie di lista dei candidati che sono particolarmente utili e che danno una base per la risoluzione di tutte le questioni chiave. Come già sottolineato, le strutture della memoria per accelerare l'aggiornamento delle valutazioni nelle mosse e ridurre lo sforzo nel trovare mosse migliori, sono importanti per sostenere il carattere aggressivo dei metodi di TS. Un precetto standard per la costruzione di liste di candidati è quello di identificare sottogruppi di mosse influenti, come un raggruppamento speciale in cui può essere dimostrata la presenza di almeno una mossa (e preferibilmente più) che sia essenziale per raggiungere una soluzione migliore. Il concetto di influenza nei segnali di TS cambia nella struttura o nella grandezza che sono misurati essenzialmente per evitare un percorso inutile.

Un tipo di strategia banale di lista di candidati è quella di campionarne casualmente alcuni da un insieme dato fino a quando ne è scelto un numero sufficientemente adeguato per averne all'interno di buoni. Ad un livello elementare, una delle più datate (ma spesso molto utili) strategie di lista di candidati è la strategia di suddivisione. Questo approccio scompone mosse

composte con l'obiettivo di isolare i "buoni componenti" che possono far parte delle mosse migliori a livello composto. Il motivo di questo approccio è che i componenti possono spesso essere valutati molto più rapidamente, e sono in genere molto meno in numero, che le mosse da questi composte. Ad esempio, "mosse di scambio" sono comunemente composte da "mosse di aggiunta" e "mosse di eliminazione", come osservato in precedenza. Il numero di queste mosse di scambio sarebbe pari in genere al prodotto dei numeri dei loro componenti di aggiunta ed eliminazione e, quindi, un approccio che cerca di valutare una frazione apprezzabile di queste mosse di scambio può richiedere molto tempo. D'altra parte, di solito è facile isolare un numero abbastanza piccolo di mosse "migliori" di aggiunta ed eliminazione, e limitare l'attenzione sulle mosse di scambio da loro composte. Questo spesso può identificare un insieme di mosse di alta qualità, nonostante il fatto che la valutazione di una mossa di scambio non può essere una semplice somma delle valutazioni dei suoi componenti. La differenza di sforzo può essere apprezzabile anche se il numero dei componenti non è grande. Migliori informazioni possono essere ottenute da valutazioni sequenziali, come avviene quando la valutazione di un componente è subordinata alla preventiva (limitata) scelta di un altro.

Un tipo un po' diverso di strategia di lista di candidati che comprende una serie interessante di varianti è la strategia di "maggior aspirazione". Questo approccio stabilisce una soglia di qualità per l'aspirazione per passare ad essere selezionate, sulla base della memoria del modello di ricerca, e prende in esame le mosse fino a trovarne una che soddisfa questa soglia. A questo punto, un ulteriore numero di mosse viene esaminato, pari ad un valore selezionato "maggior", ed è selezionata la mossa migliore globale. Per assicurare che in totale vengano esaminate né troppe né poche mosse, questa regola richiede che almeno le mosse Min e al massimo le mosse Max siano esaminate, per valori scelti di Min e Max. Quando il limite massimo di mosse Max viene raggiunto, prima di soddisfare le altre condizioni, l'approccio seleziona semplicemente la migliore delle mosse viste. I valori di Min e Max possono essere modificati in funzione del numero di mosse necessarie per soddisfare la soglia. La soglia di aspirazione per questo approccio può essere determinata in diversi modi. Per illustrarlo, durante una sequenza di mosse di miglioramento, l'aspirazione può precisare che la prossima mossa scelta deve allo stesso modo essere migliorata sia ad un livello basato su altre mosse recenti che al valore attuale obiettivo. Nel corso di una sequenza di non miglioramento l'aspirazione è in genere più bassa, ma sale verso il livello di miglioramento quando la sequenza si allunga. La qualità delle mosse attualmente all'esame può spostare la soglia, come incontrare mosse che superano in modo significativo o uniformemente cadono sotto la soglia.

Come opzione elementare, la soglia può essere semplicemente una funzione della qualità delle prime mosse Min esaminate all'attuale iterazione.

Questa strategia di maggior aspirazione include molte altre strategie come casi particolari. Ad esempio, una strategia di "primo miglioramento" risulta dall'impostazione di $Plus=0$ e dirigendo la soglia di aspirazione ad accettare mosse che si qualificano come migliori, ignorando i valori di Min e Max. Una strategia un po' più avanzata è in grado di determinare Min e Max per garantire che un numero specifico addizionale di mosse sia esaminato dopo aver soddisfatto una soglia di miglioramento. In generale nell'applicazione della strategia di *plus* aspirazione è importante assicurare che le nuove mosse siano esaminate ad ogni iterazione se non sono tra quelle appena esaminate, come partire da dove l'esame precedente si era fermato.

Un'altra applicazione utile della lista dei candidati è la strategia di lista di candidati d'élite. Questo primo approccio crea un elenco principale esaminando tutte (o un numero relativamente elevato di) mosse, selezionando le migliori k incontrate tra le mosse, dove k è un parametro di processo. Queste mosse possono essere identificate da un'altra strategia, quale la strategia di suddivisione. Poi ad ogni successiva iterazione la mossa migliore corrente è scelta dalla lista principale per essere eliminata, continuando fino a quando ci si sposta scendendo sotto una soglia di determinata qualità, o fino a un determinato numero di iterazioni trascorse. Poi viene costruita una nuova lista principale e il processo si ripete. La valutazione e la precisa identità di un movimento determinato sulla lista deve essere adeguatamente monitorato, dato che uno o entrambi possono cambiare come risultato dell'esecuzione di altre mosse dalla lista. Una tale strategia di lista di candidati d'élite può essere vantaggiosamente estesa da una variante della strategia di maggior aspirazione, permettendo ad un numero ulteriore di mosse al di fuori della lista principale l'esame ad ogni iterazione, dove quelli di sufficientemente alta qualità possono sostituire elementi della lista principale.

Può essere opportuno prendere in considerazione anche una strategia di lista di candidati a "cambio limitato", così è fornita una soluzione migliore che può essere trovata limitando il dominio delle scelte in modo che nessuna soluzione sia composta da modifiche di più di un grado limitato a qualsiasi passo. Una associazione a questo grado, espressa da una distanza metrica appropriata al contesto, viene selezionata abbastanza grande per comprendere le possibilità considerate strategicamente rilevanti. La metrica può consentire grandi cambiamenti lungo una sola dimensione, ma limitare le modifiche lungo un'altra in modo che le scelte possano essere ridotte e valutate con maggiore rapidità. Tale approccio offre particolari vantaggi come parte di una strategia di intensificazione sulla base di

decomposizione in cui la stessa scomposizione suggerisce i limiti per le modifiche della delimitazione considerata.

Un tipo di lista di candidati che è altamente sfruttabile dall'elaborazione in parallelo è un "ventaglio sequenziale" di lista candidati. L'idea di base è quella di generare qualche mossa migliore alternativa ad un determinato passaggio, e poi creare un ventaglio di percorsi di soluzioni, uno per ogni alternativa. Le diverse mosse migliori disponibili per ogni corrente sono nuovamente esaminate e solo le mosse migliori globali forniscono i nuovi flussi al passo successivo.

Nella cornice dei metodi di ricerca ad albero, un processo sequenziale come questo che opera aprendo a ventaglio è talvolta chiamato "raggio di ricerca". Una raffinatezza utile chiamata "fascio ricerca filtrata" è stata proposta e studiata da Owe Morton (1988) e sono stati suggeriti da Glover (1989) altri perfezionamenti, al di là della impostazione di ricerca ad albero. Memoria e restrizioni di TS possono essere portati avanti con ogni procedura e quindi "ereditati" nelle continuazioni esposte. In questo caso, una variazione rilevante è quella di consentire alla ricerca per ogni procedura di continuare per un certo numero di iterazioni fino a raggiungere un nuovo ottimo locale. Poi un sottoinsieme di questi possono essere selezionati e riportati. Dal momento che una soluzione scelta può essere assegnata a più di un nuovo flusso, può incarnare i diversi flussi di diverse missioni in TS, dando enfasi diversa all'intensificazione e alla diversificazione.

Nella costruzione di liste di candidati come quelle sopra citate, si nota ancora una volta che il concetto di influenza della mossa è importante per le considerazioni di più lungo termine. Così, ad esempio, i criteri di valutazione dovrebbero essere periodicamente modificati, soprattutto se non esistono mosse di miglioramento, per incoraggiare il passaggio a creare significativi cambiamenti strutturali. È richiesto il limite sul numero di mosse influenti consentite in un dato intervallo di tempo, e più in particolare sui loro effetti cumulativi interagenti, dal momento che le mosse di alta influenza possono anche essere incompatibili fra loro come base per generare soluzioni di qualità migliore.

3.4 Livelli di aspirazione

Mediante i criteri di aspirazione è introdotta la questione di ampliare delle condizioni di tabù definite ai vari livelli di restrittività, che è un importante elemento di flessibilità nel Tabu Search.

Lo stato tabù di una soluzione non è un assoluto, ma può essere annullato se certe condizioni sono soddisfatte, espresse sotto forma di livelli di aspirazione. In effetti, questi livelli di aspirazione forniscono soglie di attrattiva che decidono se le soluzioni possono essere considerate ammissibili, nonostante siano classificate tabù. Chiaramente una soluzione migliore rispetto a qualsiasi vista in precedenza merita di essere considerata accettabile. Simili criteri di aspirazione possono essere definiti su sottoinsiemi di soluzioni che appartengono a regioni o insiemi che condividono determinate caratteristiche presenti, ad esempio, in un particolare valore di una funzione od un livello di infattibilità. Uno di questi criteri di aspirazione si basa sull'individuazione di "migliori condizioni" di valori della funzione obiettivo che possono essere raggiunti con mosse che partono da determinati intervalli di valori per $f(x)$. Poi una mossa è considerata accettabile se si può raggiungere un valore nuovo migliore per l'intervallo da cui si ripartiva. L'approccio sopra descritto generalizza naturalmente intervalli di sostituzione dei valori della funzione obiettivo con altri tipi di intervalli. In questo caso è spesso preferibile prevedere che la mossa assuma un migliore valore obiettivo della funzione in relazione all'intervallo, alla fine di una mossa piuttosto che alla partenza. Ciò corrisponde più da vicino ad una restrizione tabù standard, tranne che è usata per ignorare le altre restrizioni tabù. Implicitamente corrisponde ad un tipo particolare di congiunzione.

4 MEMORIA A LUNGO TERMINE

In alcune applicazioni, i componenti di memoria a breve termine di TS sono sufficienti per produrre soluzioni di qualità molto alta. Tuttavia, in generale, TS diventa molto più forte, inserendo la memoria a più lungo termine e le sue strategie associate. Un certo numero di implementazioni di TS incorporano solo la memoria a breve termine ma sono state successivamente notevolmente migliorate con l'introduzione di componenti di memoria a lungo termine. I particolari tipi di memoria basati sulla frequenza sono fondamentali per considerazioni a più lungo termine. Questi operano per mezzo dell'introduzione di sanzioni ed incentivi determinati dal relativo arco di tempo che gli attributi sono appartenuti a soluzioni visitate, consentendo la differenziazione regionale.

Le frequenze di transizione tengono traccia di quanto spesso cambiano gli attributi, mentre le frequenze fisse tengono traccia delle relative durate per cui gli attributi si verificano in soluzioni generate. Queste memorie sono anche talvolta accompagnate da forme estese di memoria basata sulla recenza.

L'utilizzo della memoria a lungo termine, col crescere del tempo, porta più rapidamente ad una soluzione a breve termine e prima i suoi benefici diventano visibili. Spesso i suoi miglioramenti infatti cominciano a manifestarsi in una lunghezza relativamente modesta di tempo, e può consentire l'archiviazione della soluzione un po' prima di quanto sarebbe possibile altrimenti, a causa della ricerca di soluzioni di altissima qualità in un arco di tempo stretto. Per esempio sono basati su memoria a lungo termine i metodi più veloci per la memorizzazione di posti operativi e di flussi in problemi di programmazione, compresi quelli TS, con sia memoria esplicita che memoria attributiva. D'altra parte, è anche vero che la possibilità di trovare ancora soluzioni migliori in un determinato tempo cresce nel caso in cui una soluzione ottimale che non sia già stata trovata è migliorata utilizzando memoria TS a più lungo termine oltre alla memoria a breve termine.

4.1 Strategie di Intensificazione

Due componenti molto importanti nel lungo termine di TS sono le strategie di intensificazione e le strategie di diversificazione. Le strategie di intensificazione sono basate sulla scelta di modificare i vincoli e favoriscono combinazioni di mosse per il raggiungimento delle soluzioni trovate come interessanti nell'archivio della ricerca. Possono essere applicate con quartieri costruttivi e distruttivi così come di quartieri di transizione, come il riavvio di procedure che

cercano di integrare attributi ad un buon livello attuale di costruzione o distruzione (attributi subordinati precedentemente incorporati). Tali approcci hanno lavorato bene negli schemi di scelta nel tabu search probabilistico (ad esempio, Rochat e Taillard, 1995). Le strategie di intensificazione possono anche avviare un ritorno alle regioni attraenti per la ricerca in modo più accurato. Un semplice esempio di quest'ultimo tipo di approccio è mostrato nella Figura 3 (pagina seguente).

Semplice approccio di TS all'intensificazione

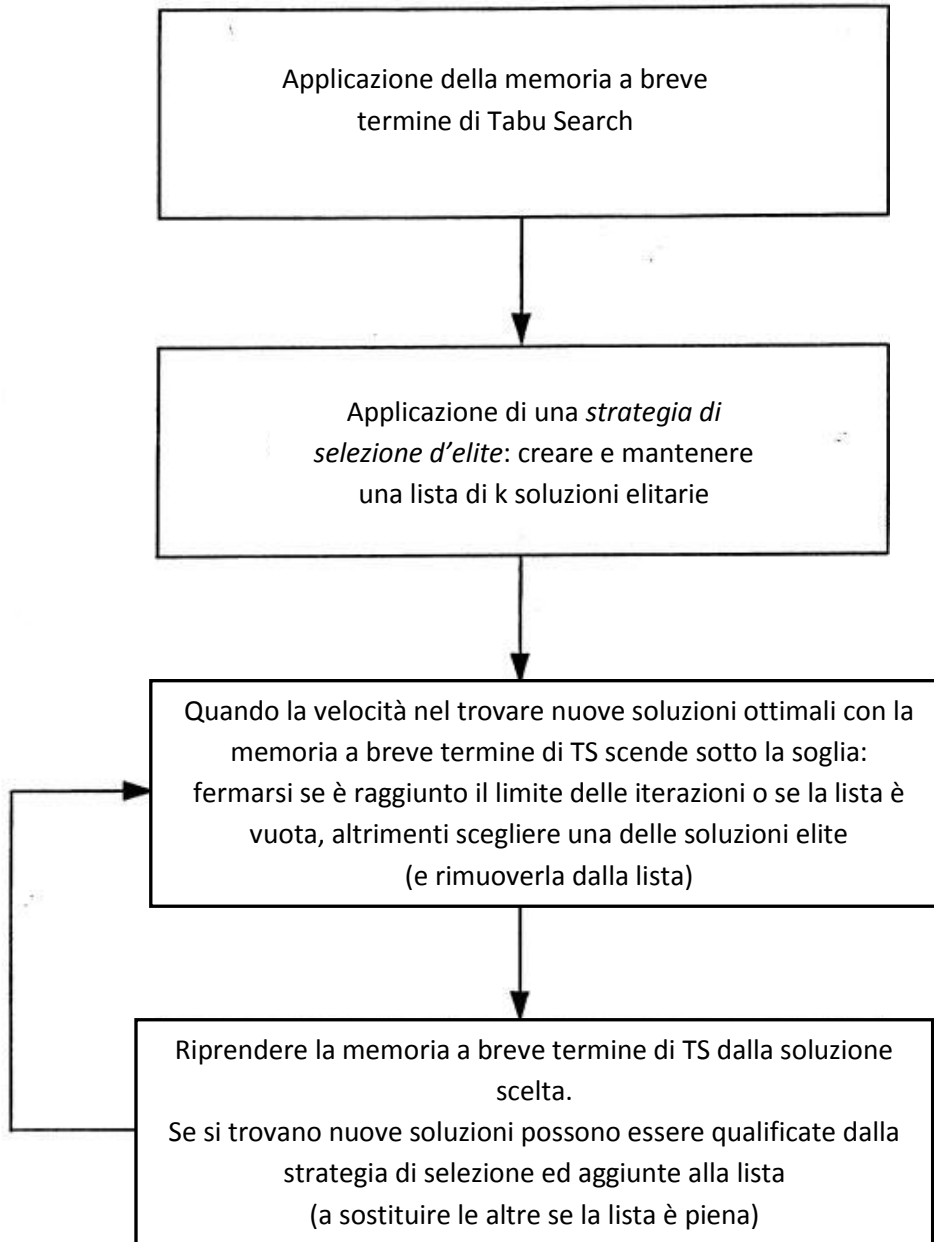


Figura 3

La strategia per la scelta di soluzioni d'elite è in corsivo nel diagramma a causa della sua importanza.

Tre varianti hanno dimostrato un discreto successo. Uno, dovuta a Voss (1993), introduce una misura per la diversificazione nell'assicurare che le soluzioni registrate differiscano le une dalle altre da un grado desiderato, e poi cancella tutta la memoria a breve termine prima di

riprendere con le migliori delle soluzioni registrate. La seconda variante, dovuta a Nowicki e Smutnicki (1993), mantiene una lista di limitata lunghezza sequenziale che aggiunge una nuova soluzione, alla fine solo se è meglio di qualsiasi vista in precedenza. L'attuale ultimo membro della lista è sempre quello scelto (e rimosso) come base per riprendere la ricerca. Tuttavia la memoria a breve termine di TS che ha accompagnato questa soluzione, inoltre, è salvata e la prima mossa proibisce anche la mossa precedentemente presa da questa soluzione in modo che sarà lanciato da un percorso di soluzione nuova. Un approccio simile, molto efficace, è stato introdotto da Barnes e Chambers (1992). La terza variante, dovuta a Xu, Chiu e Glover (1995), mantiene una lista ordinata di k soluzioni migliori, e poi, dopo un determinato numero di iterazioni, comincia dal peggior elemento di questa lista e progredisce verso il migliore. L'elemento attualmente selezionato lancia una nuova ricerca, utilizzando la ricerca probabilistica tabù come alternativa al recupero della memoria precedente. Solo un numero fisso di iterazioni è consentito al riavvio da una tale soluzione per poi risalire il prossimo, ma l'elenco continua ad essere aggiornato. Cioè, ogni volta che una soluzione migliore si trova rispetto alla peggiore che resta da essere esaminata, questa soluzione è inserita nella posizione appropriata e la peggiore soluzione che rimane è rimossa dalla lista. Questa strategia si è dimostrata molto efficace per i problemi di progettazione delle reti di telecomunicazione.

Questi approcci sono un esempio di ciò che è talvolta chiamato un "approccio ristrutturato alla mossa", riflettendo il fatto che l'insieme di mosse normali è periodicamente modificato per consentire un salto diretto ad una soluzione al di fuori del quartiere consueto. Una forma correlata di questo approccio tiene traccia dei migliori vicini nell'intorno non visitati (da quelli esaminati nelle liste candidati), con una disposizione per limitare l'attenzione a specifici tipi di soluzioni, come i vicini a ottimi locali o vicini a soluzioni visitate su gradini immediatamente prima di raggiungere tali ottimi locali (Glover 1990). Anche se questa strategia del "vicino non visitato" sembra essere trascurata, è da notare che nelle relative strategie precedentemente indicate sono fornite soluzioni di qualità molto elevata. Ad esempio, lo studio di Vaessens, Aarts e Lenstra (1994) e i documenti sull'approccio di Nowicki e Smutnicki (1993) sono insuperabili per la risoluzione di problemi di pianificazione.

Un altro tipo di approccio è l'"intensificazione della decomposizione", in cui possono essere imposte restrizioni sulle parti della struttura del problema o della soluzione al fine di generare una forma di decomposizione che permetta una messa a fuoco più concentrata su altre parti della struttura. Un classico esempio è fornito dal problema del commesso viaggiatore, in cui i rami che appartengono all'intersezione di visite elite può essere "bloccato dentro" la soluzione, al fine di concentrarsi sulla loro gestione in altre parti dell'esplorazione. L'uso di intersezioni può essere visto come un esempio estremo di una strategia più generale che mira

ad identificare e vincolare i valori delle variabili fortemente determinate e coerenti. In questo approccio, le informazioni sulle frequenze tengono traccia delle variabili che ricevono particolari valori (o che si trovano in particolari portate) in sottoinsiemi di soluzioni di elite (Glover 1977). La qualità delle soluzioni in cui queste assegnazioni di valori avvengono, e l'effetto dirimpente di modificare queste assegnazioni, mostrano la misura della loro forza. Vincolare i valori delle variabili del caso può portare ad identificare ulteriori variabili che qualificano per essere allo stesso modo vincolate, così da imprimere un elemento ricorsivo all'approccio. L'effetto complessivo può essere paragonato alla creazione di un'implosione combinatoria di possibilità, in analogia alla nozione di invertire un'esplosione combinatoria, in quanto variabili discrete vincolanti, come fissarla temporaneamente e poi rilasciandola, opera esattamente al contrario dell'aggiunta di nuove variabili discrete. Questo tipo di approccio è stato applicato efficacemente con l'intensificazione nel problema di instradamento dei veicoli (vehicle routing problem), Rochat e Taillard (1995). L'intensificazione della decomposizione comprende anche altri tipi di considerazioni strategiche, basando la decomposizione non soltanto su indicatori di forza e di coerenza, ma anche sulle opportunità, in particolare per gli elementi, di interagire produttivamente. Nel contesto dei problemi di permutazione coordinata, che possono essere convenientemente definiti in riferimento ai grafici (come in programmazione, instradamento dei veicoli e di FST), una decomposizione può essere basata sull'individuazione di sottocatene di soluzioni d'elite, dove due o più sottocatene possono essere assegnate ad un insieme comune se contengono nodi che sono "fortemente attratti" e legate con nodi di sottocatene ad altri insiemi. Un insieme di rami disgiunti della sottocatena può essere trattato da un processo di intensificazione che opera in parallelo su ogni insieme, fatto salvo che non saranno alterate le limitazioni dell'identità dei punti di fine delle sottocatene. Come risultato della decomposizione, i migliori nuovi insiemi di sottocatene possono essere riassemblati per creare una nuova soluzione. Tale processo può essere applicato a scomposizioni più alternative in forme più ampie di intensificazione di decomposizione.

4.2 Strutture di memoria basate sulla frequenza

Ancora una volta consideriamo l'impostazione del problema di ottimizzazione zero-uno, facendo riferimento a un insieme di attributi $S = \{1, \dots, s\}$ che consiste di indici di x_i variabili tra 0 ed 1. La forma di memoria di transizione per registrare il numero di cambiamenti x_i volte il

suo valore consiste semplicemente nel mantenere un contatore per x_i che viene incrementato ad ogni mossa in cui tale cambiamento si verifica. Dal momento che x_i è una variabile zero-uno, ad un richiamo rivela anche il numero di modifiche di x_i volte per ognuno dei suoi possibili valori assegnati. In situazioni più complesse, può essere utilizzata una matrice di memoria per determinare il numero di transizioni che coinvolgono le assegnazioni come $x_j=k$. Comunque, utilizzando questa memoria, le sanzioni e gli incentivi si basano su numeri relativi, anziché le cifre assolute, di transizioni, quindi richiedono che i valori registrati siano di transizione, diviso il numero totale di iterazioni (o il numero totale di transizioni).

Il salvataggio richiede uno sforzo di memoria solo poco maggiore del mantenere la memoria di transizione, approfittando della memoria basata sulla recenza memorizzata in $tabu_start(i)$. L'approccio seguito è possibilmente usato per tenere traccia del numero di soluzioni in cui $x_i=1$, consentendo in tal modo il numero di soluzioni in cui $x_i=0$ sarà dedotto da questo. Inizia con $residence(i)=0$ per ogni i , poi ogni volta che x_i cambia da 1 a 0, dopo aver aggiornato $current_iteration$ ma prima di aggiornare $tabu_start(i)$, si imposta $residence(i) = residence(i) + current_iteration - tabu_start(i)$.

Poi, durante le iterazioni quando $x_i = 0$, $residence(i)$, si deposita correttamente il numero di soluzioni in precedenza in cui $x_i=1$. Durante le iterazioni quando $x_i=1$, il valore "vero" di $residence(i)$ è il lato giusto della precedente assegnazione, ma l'aggiornamento è solo per essere fatto nei punti indicati in caso di modifiche di x_i da 1 a 0. Come con la memoria di transizione, la memoria di $residence$ dovrebbe essere tradotta in una misura relativa (dividendo per il numero totale di iterazioni, le soluzioni generate), come base per la creazione di sanzioni ed incentivi. La grandezza preferita di sanzioni e incentivi, quando non preventiva, è stabilita dalla prova empirica.

Ci sono vari modi di sfruttare la memoria basata sulla frequenza. Per illustrare alcune delle possibilità di base, possiamo considerare che divide gli attributi spostati in sei classi di frequenza, a seconda di questi attributi: (1) si verificano spesso in soluzioni buone (o ottime), (2) si verificano spesso in soluzioni povere ma raramente in soluzioni buone, (3) si verificano spesso in mosse per aggiungere l'attributo alla soluzione corrente, dove queste mosse ricevono valutazioni che sono alte, ma non "abbastanza alte" per essere scelte, (4) si verificano spesso in mosse che calano l'attributo dalla soluzione attuale, dove queste mosse allo stesso modo ricevono valutazioni non sufficientemente attraenti per essere scelte, (5) si verificano spesso nelle soluzioni effettivamente generate durante il processo di ricerca (sia buone o cattive), (6) spesso non si verificano nelle soluzioni generate.

Gli attributi di classe (1) e (2) possono essere utilizzati per sostenere gli obiettivi di intensificazione selezionando le mosse dove rispettivamente aggiungere o diminuire questi

attributi dalla soluzione. Le classi (3) e (4) combinano gli elementi di intensificazione e diversificazione di queste stesse rispettive strategie. Infine, in senso inverso, mosse che eliminano gli attributi di classe (5) e li aggiungono in classe (6) servono a sottolineare le questioni di diversificazione. L'approccio di analisi chiamato "analisi obbiettivo" può essere utilizzato per definire la soglia implicita da termini come spesso e bene, invece di ricorrere a scelte arbitrarie o sforzi di calibrazione basati su tentativi ed errori. Altri tipi di classificazioni sono naturalmente possibili, compresi quelli che coinvolgono relazioni condizionate. In tabu search l'utilizzo di memoria basata su attributi conduce naturalmente al concetto parallelo di valutazione basata su attributi, così come enunciati nella strategia di memoria a lungo termine.

4.3 Strategie di diversificazione

Le strategie di diversificazione in TS, come suggerisce il nome, sono progettate per guidare la ricerca in nuove regioni. Spesso si basano sulla modifica di norme di scelta per portare nella soluzione attributi che vengono utilizzati di rado. In alternativa, possono introdurre tali attributi parzialmente o completamente per riiniziare il processo di soluzione. Lo stesso tipo di memorie precedentemente descritte sono utili come base per tali procedure, anche se queste memorie mantengono diversi sottoinsiemi di soluzioni (generalmente più grandi) da quelli utilizzati da strategie di intensificazione. Un approccio semplice di diversificazione che mantiene una frequenza di memoria basata su tutte le soluzioni precedentemente generate, e che si è rivelato un grande successo per problemi di programmazione di macchine, è mostrato nella Figura 4 (pagina seguente).

Semplici approcci di diversificazione di TS

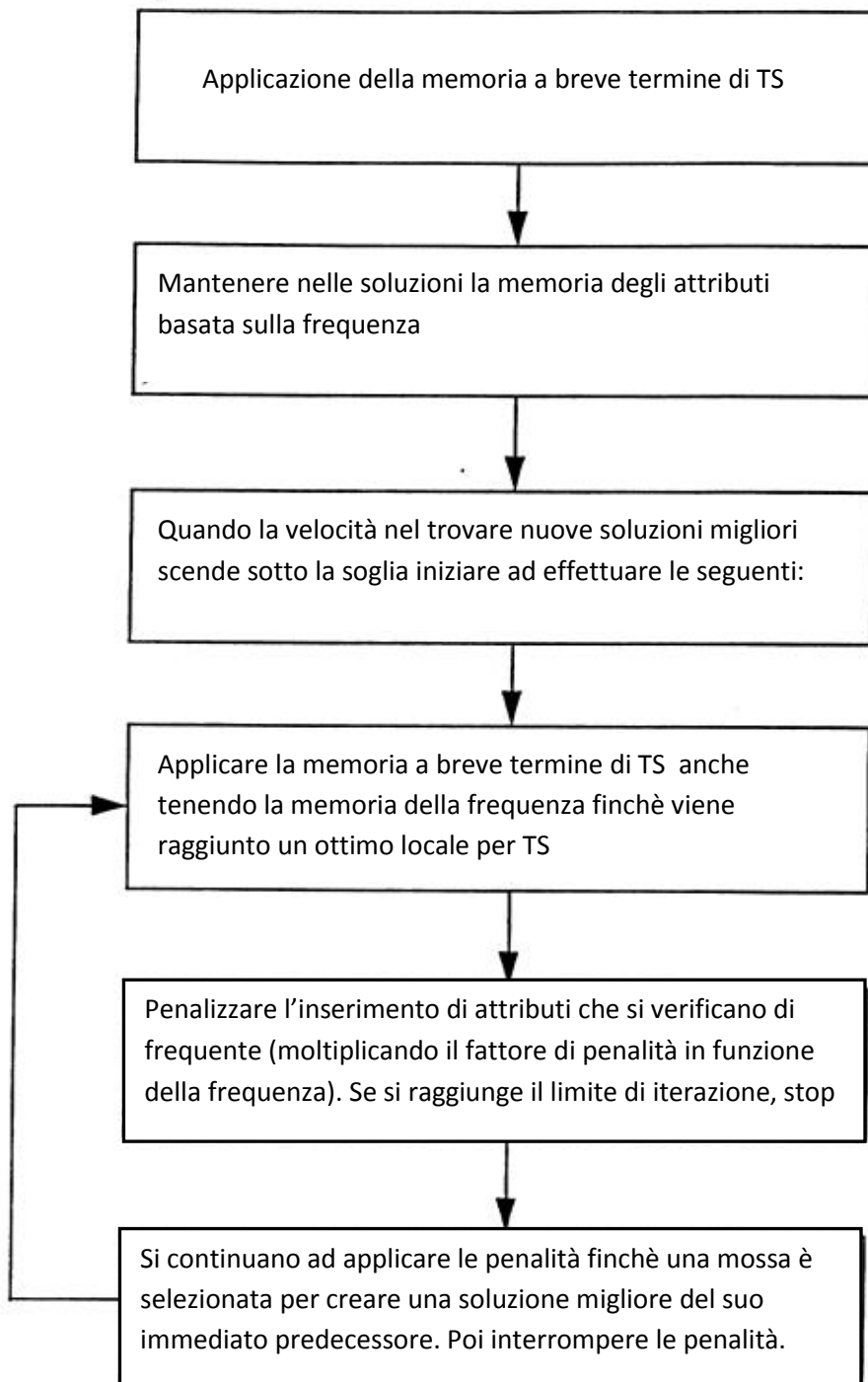
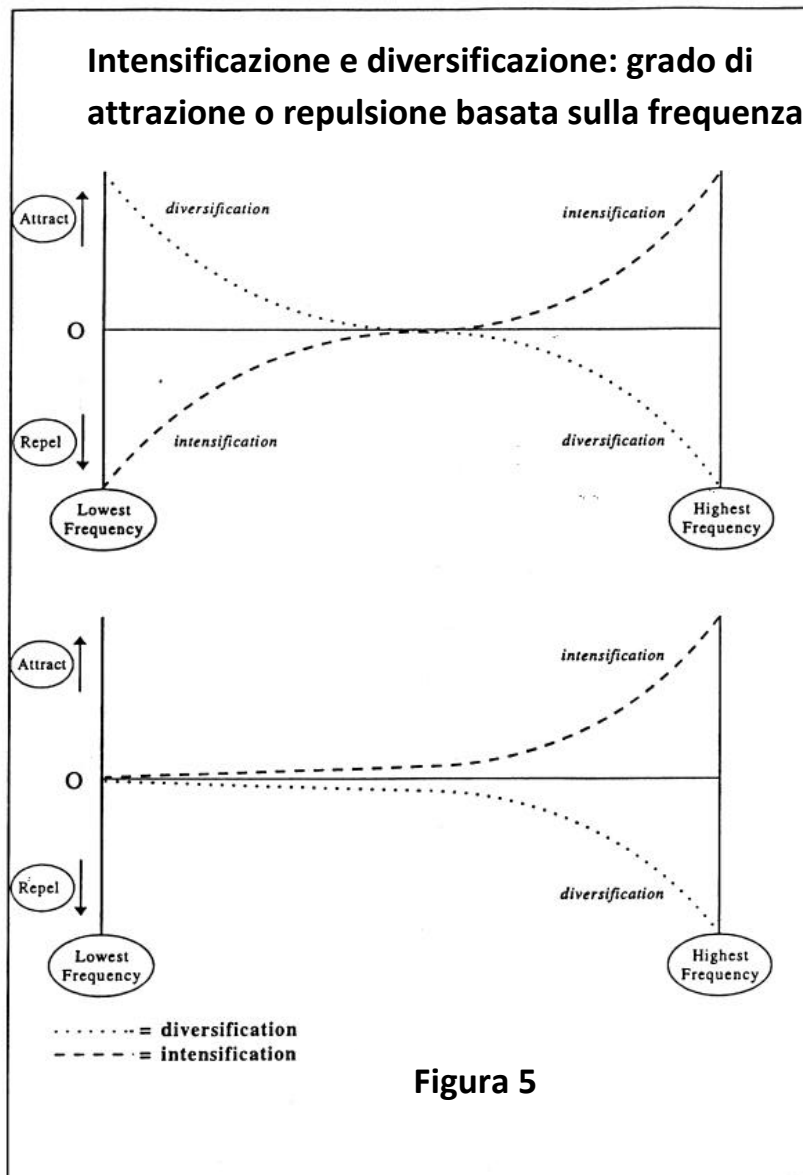


Figura 4

Con la procedura della Figura 4 sono stati raggiunti miglioramenti significativi rispetto all'applicazione della memoria a breve termine di TS (vedi Laguna e Glover 1993). Tuttavia, va sottolineato che la tempistica per l'introduzione di una diversificazione in questo approccio è

importante. La diversificazione non è applicata arbitrariamente, ma solo a ottimi locali. Inoltre, mosse migliori sono ancora selezionate per guidare il processo, oggetto di sanzioni per la diversificazione, che hanno un limitato periodo di funzionamento. L'ottimo locale di TS raggiunto da questo approccio è utilizzato come base per il lancio di una sequenza di diversificazione dei passaggi, naturalmente potrebbe essere diverso dai veri ottimi locali dalle regole TS scelte e possono escludere alcune mosse di miglioramento. Il successo di questo approccio suggerisce di introdurre una variante TS che continua sempre verso un ottimo locale vero ed una mossa una volta migliorata diventa una scelta accettabile basata su un criterio di aspirazione che viene attivato solo dopo l'esecuzione di una mossa di miglioramento. In questo approccio, fintanto che esistono ulteriori mosse di miglioramento, il criterio di aspirazione consente ad uno di loro di essere selezionato da una regola di valutazione tabù che penalizza le scelte in base al loro stato tabù, limitando l'attenzione per l'insieme di miglioramento. Una volta che un vero ottimo locale è raggiunto, il criterio di aspirazione speciale è sospeso fino a che una nuova mossa è selezionata per migliorare le regole standard del TS. Questo approccio rappresenta un'istanza di aspirazione di direzione di ricerca e può essere utilmente raffinata prendendo in considerazione le sfere di influenza (Glover e Laguna 1993). Il modo preciso in cui vengono utilizzate le memorie di frequenza a base di mettere in atto strategie di intensificazione e di diversificazione, a parte la definizione di queste memorie su diversi sottoinsiemi, fornisce un zona fertile per la ricerca. Due modelli diversi generali per sfruttare queste memorie sono illustrati nella Figura 5 (pagina seguente).



Una varietà di ulteriori alternative può essere dedotta dalla variazione naturale di tali modelli. Strategie di diversificazione che creano riavvio totale o parziale sono importanti per problemi e strutture di intorni in cui una traiettoria di soluzione può essere isolata da nuove utili alternative a meno che sia introdotto un cambiamento radicale. Forme speciali di diversificazione, in questi casi sono stati sviluppate da Hertz e de Werra (1991), Gendreau, Hertz e Laporte (1991), Soriano e Gendreau (1993), Porto e Ribeiro (1993), e Hubscher e Glover (1993).

La strategia di diversificazione può anche utilizzare una forma di memoria a lungo termine basata sull'attualità, che aumenta i risultati di permanenza in carica di attributi nella soluzione tabù. Una versione semplice di questo approccio che ha prodotto buoni risultati è mostrata nella Figura 6 (pagina seguente).

Diversificazione usando la memoria a lungo termine basata sulla recenza

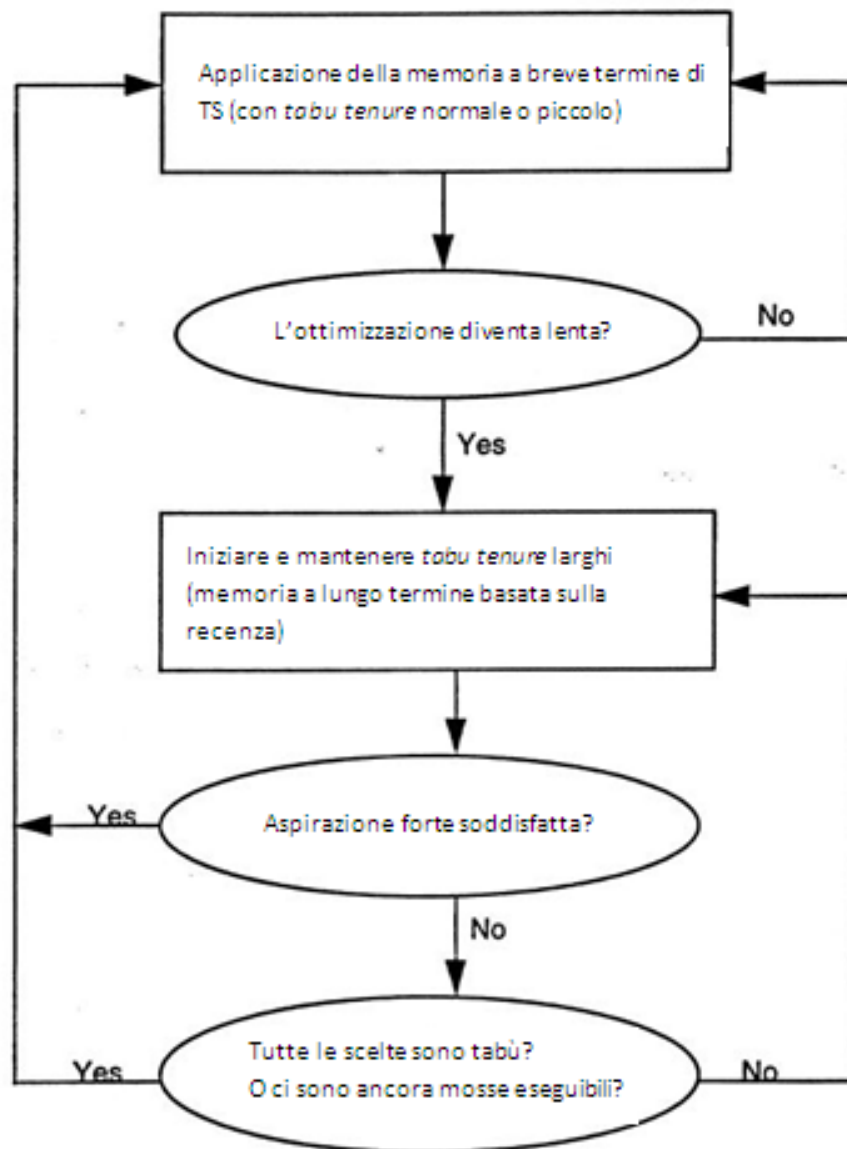


Figura 6

La ragione del successo di questo approccio è che non è diversificazione "cieca" ma incorpora implicitamente l'intensificazione. Cioè, ogni movimento continua ad essere fatto con una regola di scelta aggressiva che seleziona la migliore opzione disponibile da quelle ammissibili (ancora una volta, permettendo l'uso della strategia della lista di candidati, come esposto alla sezione 3.3). Il requisito forzato di spostare progressivamente lontano da una particolare soluzione può obbligare alcune mosse poco attraenti, ma le mosse migliori sono ancora tabù nella classe considerata. L'obiettivo può essere espresso come quello della diversificazione influente, dove l'influenza comprende il concetto di qualità. La diversificazione di questo tipo,

in un certo senso probabilistica, che riguarda l'inclusione dell'intensificazione, è una forma di forte diversificazione, sotto l'aspettativa che le soluzioni di maggiore qualità siano distribuite nello spazio di soluzione in modo che la probabilità di incontrarle sia relativamente piccola. Quindi, una soluzione che è "lontana" da un'altra, ma che è di alta qualità, è meno probabile che sia raggiunta da una serie di mosse casuali che applicano lo stesso numero di passaggi. Il concetto può essere raffinato, considerando la separazione contemporanea da più di una "soluzione di riferimento" e utilizzando il percorso di ricollegamento, concetto successivamente discusso. La determinazione dei modi efficaci per equilibrare intensificazione e diversificazione rappresentano un settore di ricerca interessante e si trovano anche al centro delle effettive implementazioni parallele di trasformazione. Dal punto di vista di TS l'obiettivo è la progettazione di modelli di comunicazione e la condivisione di informazioni tra sottoinsiemi di processori al fine di ottenere i migliori compromessi tra l'intensificazione e la diversificazione delle funzioni. In Taillard (1991), Battiti e Tecchioli (1992), Chakrapani e Skorin-Kapov (1991), Crainic, Tolosa e Gendreau (1993), e Voss (1994) sono riportati analisi generale e studi di elaborazione parallela con TS.

5 OSCILLAZIONE STRATEGICA

L'oscillazione strategica è strettamente legata alle origini del Tabu Search e fornisce un mezzo per realizzare un'interazione efficace tra l'intensificazione e la diversificazione nel medio-lungo termine. L'approccio funziona muovendosi in relazione ad un livello critico, individuato da una fase di costruzione o da un intervallo scelto di valori per una funzione.

Un tale livello critico rappresenta spesso un punto in cui il metodo normalmente si ferma. Invece di fermarsi quando questo livello viene raggiunto, tuttavia, le regole per le mosse da selezionare sono modificate, per consentire di attraversare la regione definita dal livello critico. L'approccio procede poi per una profondità determinata oltre il livello critico, e si gira. Il livello critico di nuovo si avvicina e viene attraversato, questa volta dalla direzione opposta, e il metodo viene ricavato per una nuova svolta. Il processo di affrontare e superare il livello critico da direzioni diverse crea un andamento oscillante, che dà al metodo il suo nome. Il controllo di questo comportamento è stabilito generando valutazioni modificate e regole di circolazione, a seconda della regione di navigazione e della direzione della ricerca. La possibilità di ripercorrere una traiettoria precedente è evitata dai meccanismi standard di Tabu Search.

Un semplice esempio di questo approccio si verifica nel problema dello zaino multidimensionale, in cui i valori di variabili zero-uno sono cambiati da 0 a 1 fino a raggiungere il limite di fattibilità. Il metodo continua poi nella regione fattibile utilizzando lo stesso tipo di cambiamenti, ma con una modifica da valutare. Dopo un certo numero di passi, la direzione è invertita, scegliendo le mosse che cambiano le variabili da 1 a 0. I criteri di valutazione che guidano verso il miglioramento variano a seconda che il movimento avvenga all'interno o all'esterno della regione fattibile (e se è diretta verso o lontano dal confine), accompagnata da limitazioni connesse alle modifiche ammissibili ai valori delle variabili. Le implementazioni di un approccio di questo tipo sono attribuibili a Fréville e Plateau (1986) e più recentemente da Glover e Kochenberger (1995), hanno generato soluzioni di qualità particolarmente elevata per i problemi dello zaino multidimensionale.

Un tipo un po' diverso di applicazione si verifica per problemi di teoria dei grafi in cui il livello critico rappresenta una forma desiderata di struttura del grafo, che può essere generata da aggiunte progressive (o inserzioni) di elementi base come nodi, rami o sottografi. Un tipo di approccio dell'oscillazione strategica a questo problema deriva da un processo costruttivo che introduce elementi fino a che è raggiunto il livello critico, e quindi con l'introduzione di ulteriori elementi per attraversare il confine che è definito dal livello critico. La soluzione

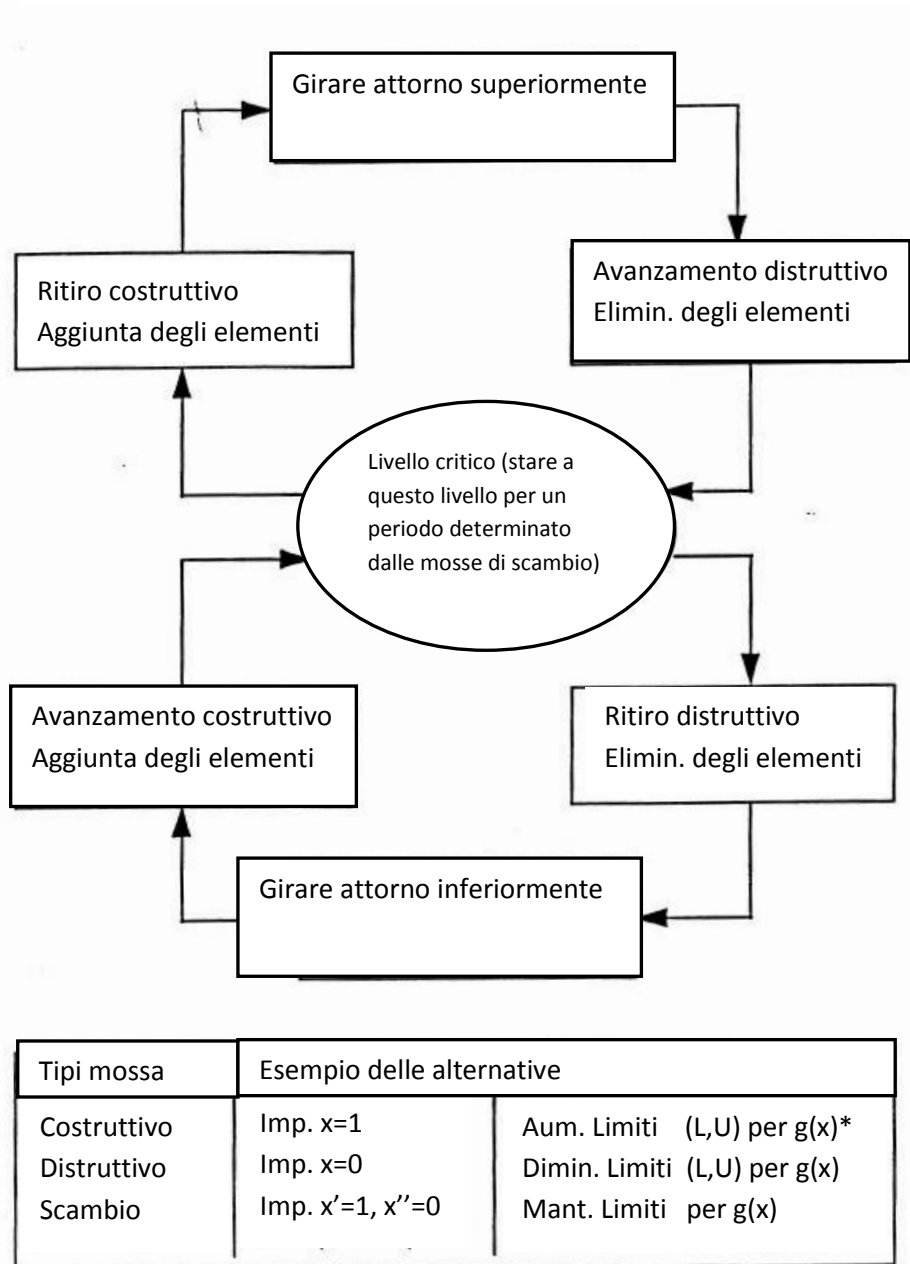
corrente può cambiare la sua struttura una volta che questo confine è attraversato, e quindi può essere richiesto un intorno diverso, producendo nuove regole per la selezione delle mosse. Le regole di nuovo cambiamento, al fine di procedere nel senso opposto, operano la rimozione di elementi fino a quando è raggiunto di nuovo il recupero della struttura che definisce il livello critico. Una simile norma che modifica in base alla direzione e la fase di ricerca sono caratteristiche tipiche dell'oscillazione strategica e forniscono una maggiore vitalità euristica. L'applicazione di norme diverse possono essere accompagnate dall'attraversamento di un confine a profondità diverse su lati diversi. Una possibilità è quella di avvicinarsi e di ritirarsi dal confine, pur rimanendo su un singolo lato, senza attraversare, cioè selezionare un passaggio di "zero profondità".

Entrambi questi esempi costituiscono un tipo strategico di oscillazione costruttiva e distruttiva, dove misure costruttive "aggiungono" elementi (o variabili dall'insieme di 1) e passaggi distruttivi "cancellano" elementi (o variabili dall'insieme di 0). Oscillazioni unilaterali sono spesso approcci costruttivi e distruttivi particolarmente rilevanti, come nel contesto di una varietà di problemi di teoria di programmazione e grafica, dove una struttura utile può essere mantenuta fino a un punto critico e poi si perde. In questi casi, il processo costruisce per il livello critico e inverte quindi l'applicazione di movimenti distruttivi. Dopo che è completata una fase costruttiva consistente in una serie di "mosse di aggiunta", la "mossa di eliminazione" più attraente per la fase distruttiva potrebbe avere poco a che fare con la sequenza in cui sono stati aggiunti elementi. Tuttavia, le strutture di memoria di TS sono ancora necessarie per assicurare le fasi alterne e non si annullano a vicenda in modo efficace.

Negli approcci di oscillazione strategica spesso è importante spendere tempo supplementare di ricerca in regioni vicine al livello critico e soprattutto trascorrere del tempo al livello critico stesso. Questo può essere fatto inducendo una sequenza di oscillazioni strette attorno al livello critico come preludio ad ogni più grande oscillazione che procede ad una profondità maggiore. In alternativa, se è consentito maggiore sforzo per la valutazione e l'esecuzione di ogni mossa, il metodo può usare "mosse di scambio" (intese in senso ampio) per soggiornare in questo livello critico per periodi più lunghi. Una scelta semplice, per esempio, è quella di usare le mosse di scambio per procedere ad un ottimo locale ogni volta che viene raggiunto il livello critico. Una strategia nell'applicare allo stesso modo gli scambi a ulteriori livelli è suggerito da un principio di prossima ottimalità, la quale afferma approssimativamente che buone costruzioni ad un certo livello sono probabilmente vicine a buone costruzioni ad un altro. La versione semplice di una oscillazione strategica di forma costruttiva e distruttiva è illustrata in Figura 7 (pagina seguente). Come si osserva nella tabella che accompagna la Figura 7, l'oscillazione può operare anche aumentando e diminuendo i limiti per una funzione $g(x)$. Tale

approccio è stato la base per una serie di applicazioni efficaci, dove $g(x)$ ha rappresentato elementi quali le assegnazioni della forza lavoro, valori di funzione obiettivo, e di livelli di fattibilità o infattibilità, per guidare la ricerca con sonda a diverse profondità nelle regioni associate. Quando i livelli si riferiscono ai gradi di fattibilità ed infattibilità, $g(x)$ è un vettore con valori di funzione associati ad un insieme di problemi di vincoli, che può riassumere per esempio $g(x)$. In questo caso, il controllo della ricerca per g di delimitazione (x) può essere visto come la manipolazione di una parametrizzazione del vincolo selezionato insieme. Una valida alternativa è spesso preferita a fare $g(x)$ una funzione lagrangiana o surrogata con vincolo di pena, evitando funzioni valutate vettoriali e permettendo compromessi tra i gradi di violazione di vincoli di diversi componenti. Approcci che incarnano queste idee si possono trovare, per esempio, in Fréville e Plateau (1986), Gendreau, Hertz e Laporte (1993), Kelly, Golden e Assad (1993), Osman (1993), Osman e Christofides (1993), Rochat e Semeraro (1993), e Voss (1993).

Oscillazione strategica usando mosse costruttive o distruttive



* $g(x)$ può essere una funzione valutata scalare o vettoriale della soluzione x

Figura 7

5.1 Strutture di memoria dell'oscillazione strategica

L'oscillazione strategica offre la possibilità di fare un particolare uso della memoria basata sulla frequenza sia a breve che a lungo termine. Per illustrare, sia $A(\text{current_iteration})$ a denotare un vettore zero-uno il cui componente ha il valore 1 se l'attributo j è presente nella soluzione

corrente ed ha il valore 0 altrimenti. Il vettore A può essere trattato come se fosse lo stesso del vettore soluzione di problemi zero-uno, anche se implicitamente lo è due volte più grande, dato che $x_j=0$ è un attributo diverso da $x_j=1$. Questo significa che le norme per l'esercizio sul pieno A devono essere reinterpretate per l'esercizio sulla forma condensata di A . La somma dei vettori A in tutte le t iterazioni più recenti fornisce una semplice memoria che unisce considerazioni di recenza e frequenza. Per mantenere la somma si richiede di ricordare $A(k)$, per k che varia nel corso delle ultime t iterazioni. Allora il vettore somma A^* può essere aggiornato facilmente dal calcolo incrementale

$$A^* = A^* + A(\text{current_iteration}) - A(\text{current_iteration} - t + 1).$$

Le misure di frequenza associate, come sottolineato in precedenza, devono essere normalizzate, in questo caso dividendo di A^* il valore di t . Una forma a lungo termine di A^* non richiede la memorizzazione dei vettori $A(k)$, ma tiene semplicemente una somma in esecuzione. A^* può anche essere gestito dal livellamento esponenziale.

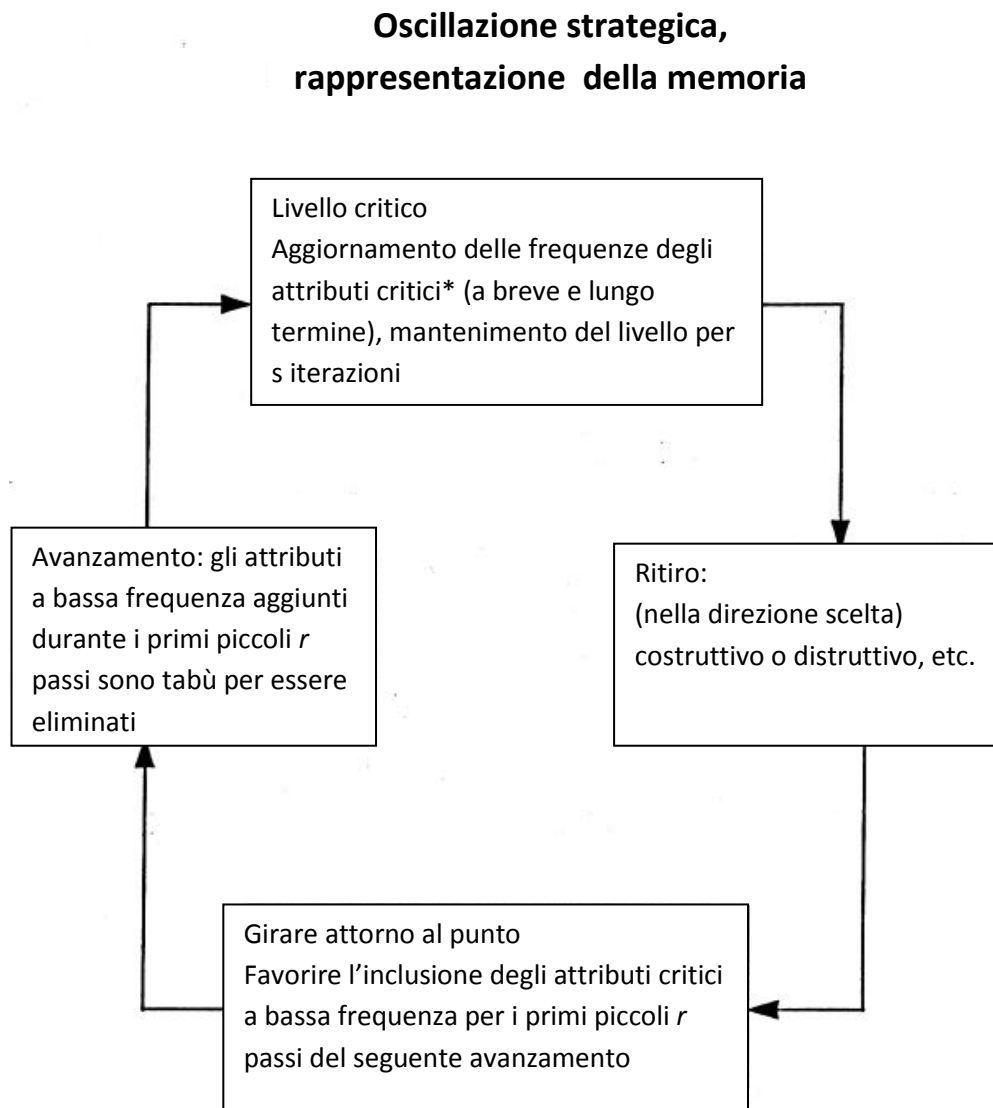
Tale memoria basata sulla frequenza è utile nell'oscillazione strategica per la seguente osservazione.

Invece di usare una consueta memoria basata sulla recenza ad ogni passo di un modello oscillante, un maggiore risultato è dato dalla flessibilità ignorando le restrizioni tabù fino a raggiungere il punto di svolta. A questo punto, assumendo una regola di scelta si applica l'introduzione di un attributo che non era contenuto in una soluzione recente al livello critico. Se questo attributo è mantenuto in soluzione rendendolo tabù per essere eliminato, poi alla fine il raggiungimento del livello critico della soluzione sarà diverso da quelli visti sopra l'ultimo orizzonte dell'iterazione t . Così, invece di aggiornare A^* ad ogni passo, l'aggiornamento viene fatto solo per il livello critico delle soluzioni, allo stesso tempo migliorando la flessibilità nel fare delle scelte.

In generale, si verifica la possibilità che non esista l'attributo che permette che tale processo sia attuato nella forma indicata. Ciò significa che ogni attributo può già avere una voce positiva associata ad A^* . Così, al punto di girare intorno, la regola sta invece nella scelta di una mossa che introduca gli attributi che sono meno utilizzati di frequente. Nota, "raramente utilizzata" può significare sia "raramente presente" sia "di rado assente", a seconda della direzione della corrente di oscillazione.

Per una maggiore diversificazione, questa regola può essere applicata per r passaggi dopo aver raggiunto il punto di giro. Normalmente r dovrebbe essere un numero ridotto, ad esempio, con un valore di base di 1 o 2, che viene periodicamente aumentato in un modello standard di diversificazione. Passando da un A^* a breve termine ad un A^* a lungo termine si crea un effetto di diversificazione globale.

Questo tipo di memoria si è rivelata particolarmente efficace per risolvere lo zaino multidimensionale e relativi problemi, soprattutto quando si utilizzano le regole scelte basate su vincoli di valutazioni surrogate (Glover e Kochenberger, 1995). Un modello di questo approccio è dato in Figura 8 .



- * Per la parte selezionata delle iterazioni a livello critico: per prime e migliori soluzioni del blocco corrente

Figura 8

L'approccio di Figura 8 non è simmetrico. Una forma alternativa di controllo è quella di cercare immediatamente di introdurre un attributo a bassa frequenza in uscita dalla struttura di livello critico, per aumentare la probabilità che la soluzione al turno successivo nel giro non duplichi

una soluzione già visitata in quel punto. Tale controllo può altresì valorizzare la diversità, anche se la sovrapposizione di compiti nei giri è già inibita partendo da diverse soluzioni a livello critico e quando tali duplicazioni tuttavia si verificano possono non sempre essere indesiderabili.

6 PERCORSO DI RICOLLEGAMENTO

Un'utile integrazione delle strategie di intensificazione e diversificazione avviene nel approccio chiamato percorso di ricollegamento (Glover, 1989). Questo approccio genera nuove soluzioni, esplorando traiettorie che collegano soluzioni èlite partendo da una di queste soluzioni, chiamate soluzioni di avvio e generando un percorso nello spazio dell'intorno che porta verso le altre soluzioni, chiamate soluzione guida. Questo si ottiene selezionando mosse che introducono attributi contenuti nelle soluzioni guida. Come si vedrà, la soluzione di avvio può essere anche una soluzione "null" o "più specifica" quando vengono utilizzati intorno costruttivi e distruttivi.

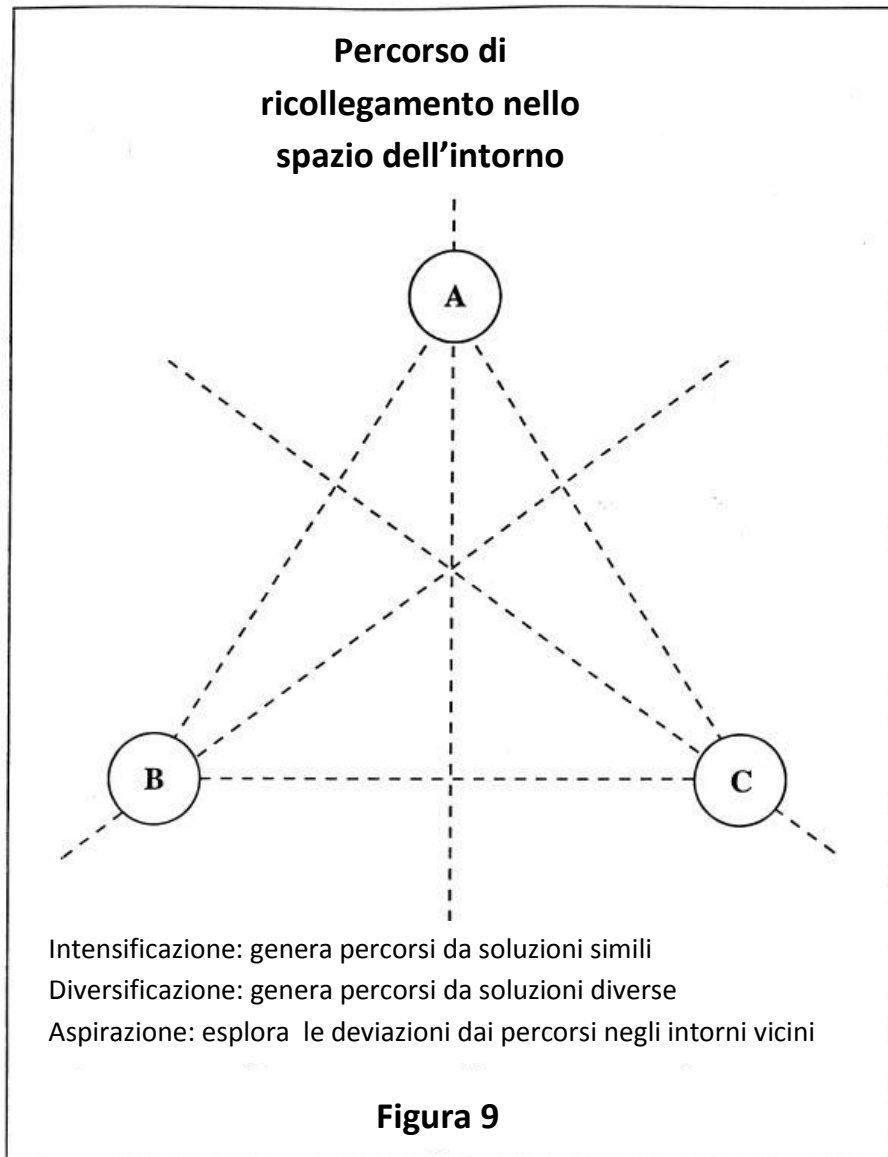
L'approccio può essere visto come un esempio estremo (altamente concentrato) di una strategia che mira a includere gli attributi di soluzioni di alta qualità, attraverso la creazione di incentivi per favorire questi attributi nelle mosse selezionate. Tuttavia, invece di utilizzare un incentivo, che si limita a sollecitare l'inserimento di tali attributi, il percorso di ricollegamento approccia subordinando tutte le altre considerazioni a raggiungere l'obiettivo di scegliere mosse che introducano le caratteristiche delle soluzioni di guida, al fine di creare una "buona composizione di attributi" nella soluzione corrente. La composizione ad ogni passo è determinata dalla scelta della mossa migliore, utilizzando criteri di scelta consueti, dall'insieme limitato di mosse che incorporano un massimo numero (o un valore massimo ponderato) degli attributi delle soluzioni di guida. Come in altre applicazioni di TS, i criteri di aspirazione possono ignorare questa restrizione per permettere ad altre mosse di particolare alta qualità di essere considerate.

In particolare per identificare un insieme di uno o più soluzioni èlite per orientare il percorso di una soluzione proposta agli attributi di queste soluzioni guida, sono assegnati i pesi preventivi come stimolo da selezionare. Pesi più grandi sono assegnati agli attributi che si verificano in numero maggiore di soluzioni guida, prevenzione in modo da poter dare maggiore enfasi alle soluzioni con qualità superiore o con funzioni speciali, ad esempio complementari a quelle della soluzione che ha avviato la nuova traiettoria.

Più in generale, non è necessario per un attributo che si verifichi in una soluzione di guida al fine di avere una posizione privilegiata. In alcune impostazioni gli attributi possono condividere gradi di somiglianza, e in questo caso può essere utile per vedere una soluzione vettore per fornire "punteggi" a favorire o scoraggiare particolari attributi (Glover, 1991). In genere, solo forme più forti di criteri di aspirazione sono autorizzate a superare questo tipo di regola scelta.

In una data collezione di soluzioni di èlite, il ruolo di avviare e guidare soluzioni può essere alternato. Cioè un insieme di soluzioni attuali possono essere generate contemporaneamente, estendendo percorsi diversi, e permettendo ad una soluzione di apertura di essere sostituita (come una soluzione di guida per gli altri) ogni volta che la soluzione corrente soddisfa un criterio di aspirazione sufficientemente forte. Poiché i loro ruoli sono intercambiabili, l'avvio e soluzioni di guida sono chiamati collettivamente "soluzioni di riferimento". Una forma idealizzata di tale processo è mostrata nella Figura 9 (pagina seguente). La collezione scelta di soluzioni di riferimento è costituita da tre membri, A, B e C. Sono tracciati in modo da consentire a ciascuno di servire come soluzione d'avvio e consentendo ad una od entrambe le altre due soluzioni di operare come guida alle soluzioni.

Soluzioni intermedie che si incontrano lungo i sentieri non sono visualizzate. La rappresentazione dei percorsi come linee dritte è semplificata, dato che scegliendo tra mosse disponibili in un intorno in genere si può ottenere una traiettoria molto più complessa.



Come indica la Figura 9 la continuazione del percorso è consentita al di là di ogni soluzione di avvio/guida. Tale proroga può essere compiuta penalizzando l'inserimento di attributi eliminati nel corso di una traiettoria, compresi gli attributi che guidano soluzioni che possono essere costrette a essere eliminate al fine di proseguire il percorso. Una soluzione di avvio può anche essere respinta dalle soluzioni guida penalizzando l'inclusione dei loro attributi fin dall'inizio. Varianti TS probabilistiche operano nell'impostazione del percorso di ricollegamento, come in altri, traducendo valutazioni di regole deterministiche in probabilità di selezione, fortemente orientata a favorire maggiori valutazioni.

Regioni promettenti possono essere ricercate in modo più approfondito nel percorso di ricollegamento modificando i pesi relativi agli attributi delle soluzioni guida e alterando la polarizzazione associata alla qualità della soluzione e le caratteristiche della soluzione

selezionata. La Figura 10 descrive il tipo di variazione che può derivare, in cui il punto X rappresenta una soluzione di avvio ed i punti A, B e C, rappresentano soluzioni di guida. Per le scelte appropriate dei punti di riferimento, e quartieri per la generazione di percorsi da questi, il principio suggerisce che ulteriori punti èlite hanno probabilità di essere trovati nelle regioni attraversate da percorsi, su lancio di nuove ricerche di alta qualità segnati su questi percorsi. La prova che gli spazi di soluzione combinatoria spesso hanno tipologie che possono essere utilmente sfruttate da un tale approccio è fornito dai risultati di Moscato (1993), Tinetti (1994), e Nowicki e Smutnicki (1993).

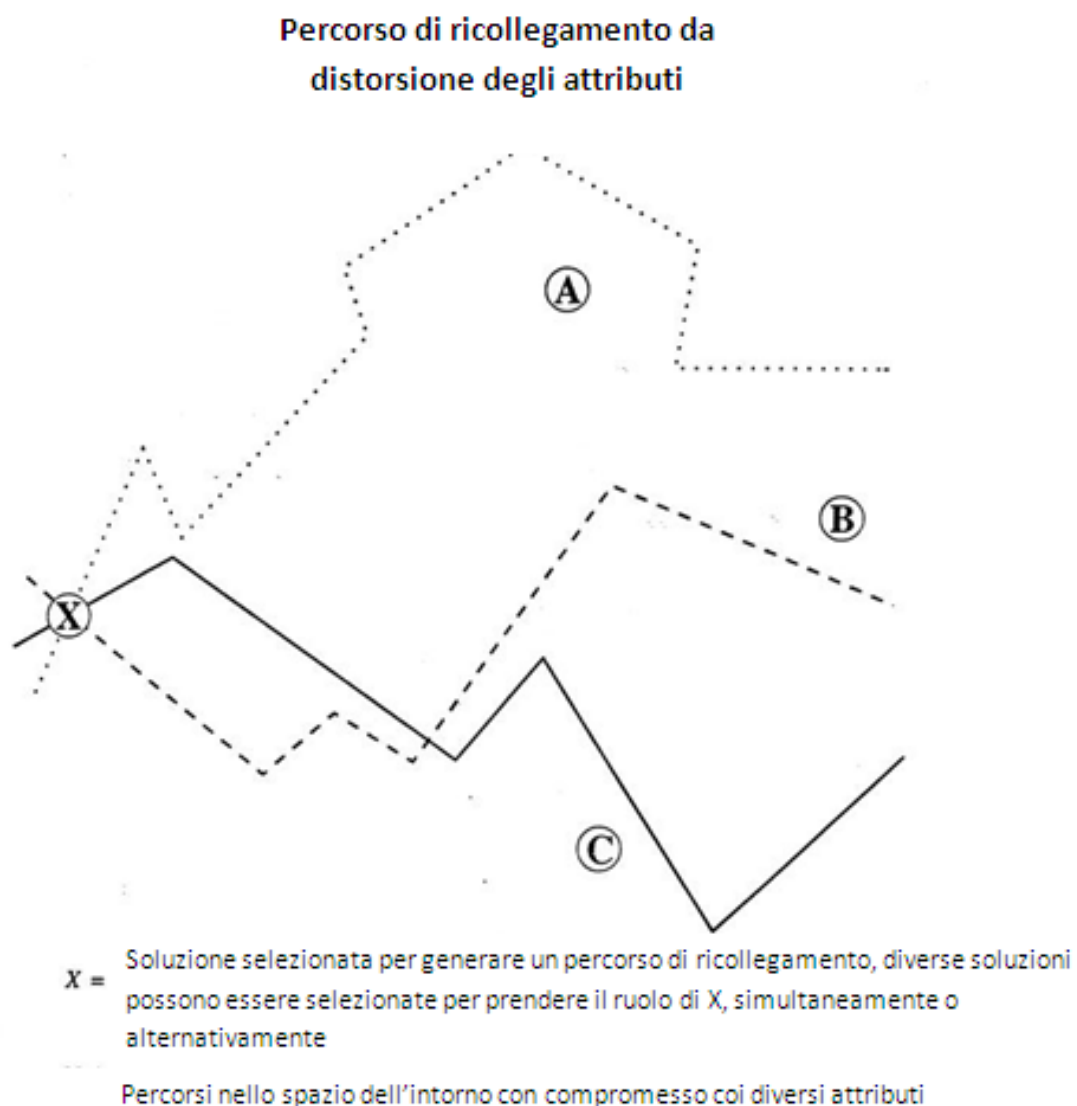


Figura 10

6.1 Considerazioni sul Percorso di ricollegamento

Le strategie del percorso di ricollegamento in TS possono occasionalmente dare profitto impiegando diversi intorni e diverse definizioni di attributi da quelle usate da euristici per la generazione di soluzioni di riferimento. Per esempio, a volte è conveniente usare un intorno costruttivo per il percorso di ricollegamento che genera una sequenza di posti di lavoro per trasformarli in macchine specifiche. In questo caso una soluzione d'avvio d'élite può essere utilizzata per dare inizio ad una costruzione parziale, specificando gli attributi particolari (come ad come posti di lavoro, in particolare, le posizioni in sequenza relativa o assoluta) come base per le restanti fasi costruttive.

I commenti su intorni costruttivi in questa sezione possono anche essere facilmente resi applicabili ai distruttivi, quando una soluzione iniziale è "sovraccarica" con gli attributi donati da soluzioni guida e tali attributi sono progressivamente spogliati di distanza o modificati fino a raggiungere un insieme con una composizione adeguata. Quando la via di ricollegamento si basa sui quartieri costruttivi, la soluzione di guida (s) fornisce le relazioni tra attributi che danno le opzioni per le successive fasi di costruzione. Come estremo, una piena costruzione per essere prodotta rende la soluzione di avvio una soluzione nulla. L'estremo distruttivo inizia da una "serie completa" di elementi di soluzione. Approcci costruttivi e distruttivi producono solo una soluzione nuova singola, piuttosto che una sequenza di soluzioni, su ogni "percorso" che porta dalla soluzione di apertura verso gli altri. In questo caso il percorso non potrà mai raggiungere gli altri a meno che sia utilizzato un quartiere di transizione per estendere il quartiere costruttivo. Una caratterizzazione di tali processi e le regole illustrative della loro esecuzione sono indicati in Glover (1991).

L'intorno costruttivo può essere visto come un caso speciale di "ripristino di fattibilità" degli intorni, dal momento che un valore nullo o parzialmente costruito di soluzione non soddisfa tutte le condizioni per qualificarsi come fattibile. Una varietà di metodi studiati per soluzioni di ripristino di fattibilità impraticabile, come esemplificato nei metodi di flusso in aumento nei problemi di rete, sono i metodi di eliminazione di sottovisite nel problema del commesso viaggiatore e nei problemi di instradamento di veicoli, alternando le procedure di catena in grado con limitazioni nei problemi di sottografi e metodi di copertura di incremento e decremento del valore e problemi di zaino multidimensionale.

L'utilizzo di intorni che permettono limitate forme di infattibilità che saranno generate, e quindi utilizzando gli associati per rimuovere queste infattibilità, forniscono una forma di percorso di ricollegamento con utili caratteristiche di diversificazione. Dopo l'introduzione di

ulteriori quartieri di transizione, con la capacità di generare soluzioni successive con mescolamento di attributi cambiati, il meccanismo del percorso di ricollegamento dà anche un modo per passare attraverso le regioni non fattibili. L'applicazione di tali processi in un quadro probabilistico di TS, traduce le valutazioni da regole deterministiche in probabilità di selezione, offrendo ulteriori opportunità per la variazione.

Una sintesi dei componenti del percorso di ricollegamento che incarna queste idee (in forma abbreviata) è riportato nel seguente schema.

SINTESI DEL PERCORSO DI RICOLLEGAMENTO

Fase 1. Identificare la struttura dell'intorno e la soluzione per i relativi attributi del percorso di ricollegamento (possibilmente diversi da quelli di altre strategie di TS applicate al problema).

Fase 2. Selezionare un insieme di due o più soluzioni di riferimento e individuare quali membri serviranno come la soluzione di avvio e la soluzione di guida (s). Per un quartiere costruttivo, identificare la parte della soluzione di apertura, forse nulla, per avviare la costruzione.

Fase 3. Passare ad avviare verso la soluzione (o oltre) la soluzione di guida (s), generando una o più soluzioni intermedie come candidati per avviare il problema successivo a sforzi per la soluzione. Se la prima fase di questo passaggio crea una soluzione non fattibile, applicare una seconda fase associata ad un intorno ammissibile del ripristino.

Il percorso di ricollegamento deriva da un gruppo base di approcci chiamati di "ricerca di dispersione", che generano nuove soluzioni con la creazione di modifiche di combinazioni lineari dei punti di riferimento.

I punti di riferimento per la ricerca di dispersione, come per il percorso ricollegamento, costituiti da soluzioni elite prodotti da processi di ricerca e le migliori soluzioni combinate sono utilizzati per ri-avviare i processi in un ciclo ripetitivo. Da un certo punto di vista, la modifica di combinazioni lineari prodotta dalla dispersione di ricerca può essere vista come generatrice di percorsi di spazio vettoriale euclideo. Una tale visione conduce per naturale estensione della nozione di spazio euclideo alla sostituzione con uno spazio di intorno, così fornendo la base per l'approccio al percorso ricollegamento.

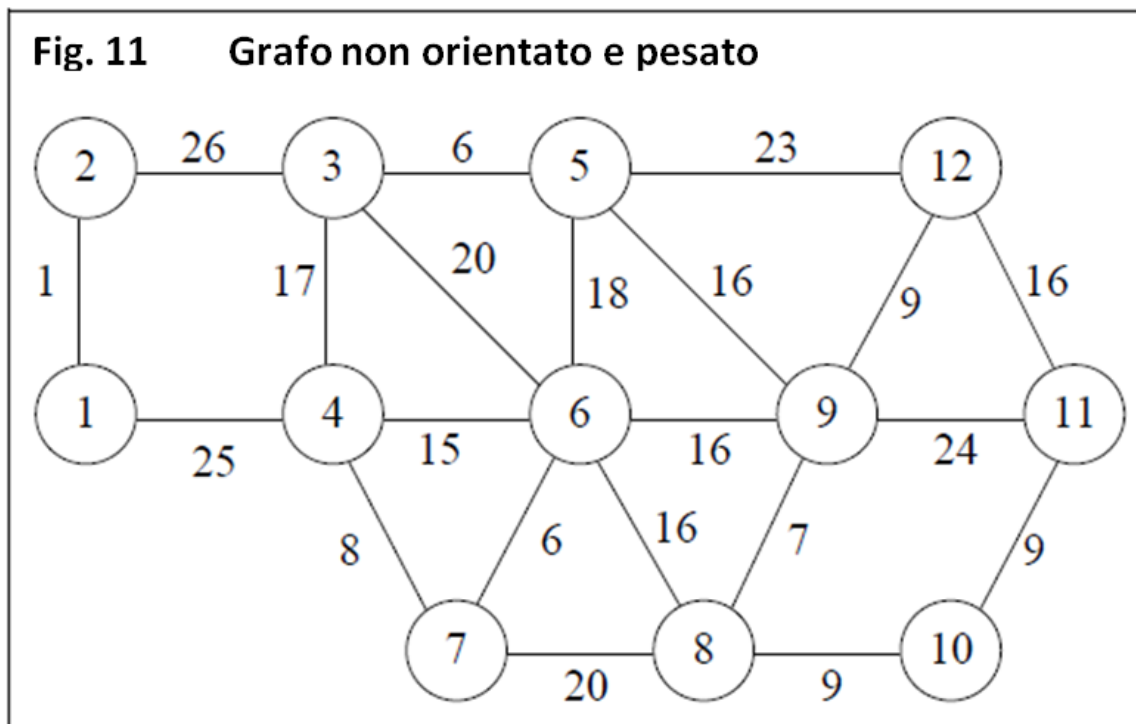
Per analogia inversa, le soluzioni prodotte dal percorso di ricollegamento possono essere viste come "combinazioni" delle loro soluzioni di riferimento. Questo fornisce un collegamento interessante tra le proposte di Tabu Search e le proposte di algoritmi genetici. In realtà, molti "operatori di crossover" di recente sviluppo in strategie di questi algoritmi, senza alcuna apparente relazione tra di loro, si può dimostrare che sorgono come istanze del percorso di ricollegamento, limitando l'attenzione su due punti di riferimento (presi come genitori

nell'algoritmo genetico) e da sostituzione della selezione strategica con una randomizzazione di fiducia.

Il percorso di ricollegamento, in comune con l'oscillazione strategica, dà un fondamento naturale per lo sviluppo di strategie di intensificazione e diversificazione. Tipicamente strategie di intensificazione in tali applicazioni scelgono le soluzioni di riferimento da soluzioni elite che si trovano in una regione in comune o che presentano caratteristiche comuni. Allo stesso modo, le strategie di diversificazione in base al percorso di ricollegamento caratteristicamente selezionano le soluzioni di riferimento che provengono da regioni diverse o che presentano caratteristiche contrastanti. Strategie di diversificazione possono anche porre maggiormente l'accento su sentieri che vanno oltre i punti di riferimento. Collezioni di punti di riferimento che incarnano tali condizioni possono essere utilmente determinati da metodi di raggruppamento. Queste forme alternative di percorso di ricollegamento offrono anche una base conveniente per l'elaborazione parallela, che contribuisce a compromessi nell'approccio all'integrazione di intensificazione diversificazione nella progettazione di processi paralleli di soluzione generale.

7 ESEMPIO APPLICATIVO: problema dell'albero con k-rami minimizzato

Il problema dell'albero-k minimizzato cerca un albero composto da k archi in un grafo in modo che la somma dei pesi di questi rami sia minimo. Un esempio di questo problema è dato in Figura 11, dove sono mostrati i nodi come cerchi numerati ed i rami sono mostrati come linee che uniscono le coppie di nodi. I pesi dei rami sono indicati come i numeri relativi a queste linee. Un albero è un insieme di archi che non contiene cicli, cioè, che non contiene i percorsi che iniziano e terminano in corrispondenza del nodo stesso (senza ripercorrere alcun ramo).



Si supponga che l'esecuzione delle "mosse" sia definita dallo scambio di rami, come ho successivamente descritto, e che sia utilizzata una determinata procedura per trovare una soluzione di partenza.

La costruzione inizia scegliendo il ramo (i, j) con il minor peso nel grafico, dove i e j sono gli indici dei nodi che sono i punti finali del ramo. I $k-1$ rami rimanenti sono scelti successivamente per ridurre al minimo l'aumento di peso totale ad ogni passo, in cui i rami considerati soddisfano esattamente un nodo da quelle che sono le estremità dei rami precedentemente scelti. Per $k = 4$, la costruzione "greedy" esegue le operazioni indicate nella Tabella 12 (pagina seguente).

Un algoritmo greedy è un algoritmo che cerca di ottenere una soluzione ottima da un punto di vista globale attraverso la scelta ad ogni passo locale della soluzione più *golosa*, aggressiva o avida, a seconda della traduzione preferita del termine *greedy* dall'inglese.

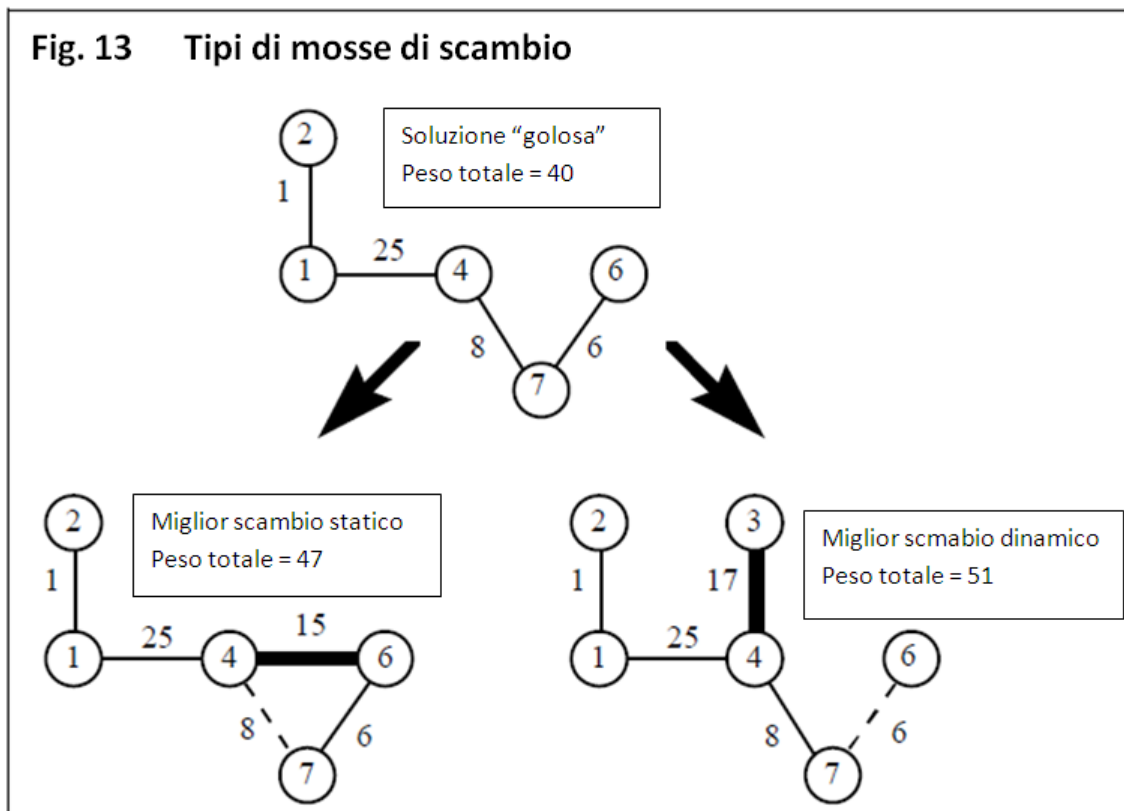
Tab. 12 Costruzione “golosa”			
Passo	Candidati	Selezione	Peso totale
1	(1,2)	(1,2)	1
2	(1,4), (2,3)	(1,4)	26
3	(2,3), (3,4), (4,6), (4,7)	(4,7)	34
4	(2,3), (3,4), (4,6), (6,7), (7,8)	(6,7)	40

La costruzione inizia con la scelta del ramo (1,2) con un peso di 1, il più piccolo peso di ogni arco del grafo. Dopo questa selezione, i rami candidati sono quelli che collegano i nodi dell'albero corrente parziale con i nodi non nella struttura, per esempio, i rami (1,4) e (2,3). Il ramo (1,4), che riduce al minimo l'aumento di peso, è scelto per essere parte della soluzione parziale. Il resto delle selezioni segue la stessa logica e la costruzione termina quando l'albero è composto da 4 rami (cioè, il valore di k). La soluzione iniziale in questo caso particolare ha un peso totale di 40.

Il meccanismo di scambio, che viene utilizzato da questo punto in poi, sostituisce un ramo selezionato nella struttura con un altro ramo selezionato al di fuori dell'albero, soggetto a richiedere che il sottografo risultante sia ancora un albero. In realtà ci sono due tipi di tale scambio di rami, uno che mantiene i nodi attuali della struttura invariata (statico) e uno che si traduce nella sostituzione di un nodo dell'albero con un nuovo nodo (dinamico). La Figura 13 (pagina seguente) illustra il migliore scambio di ogni tipo che può essere effettuato a partire dalla soluzione. Il ramo aggiunto in ogni caso è indicato da una linea continua e il ramo eliminato è indicato da una linea tratteggiata.

La mossa migliore di entrambi i tipi è lo scambio statico di Figura 13, dove a scopo illustrativo sto attualmente definendo solo in termini di variazione del valore della funzione obiettivo. Dal momento che questi migliori risultati vanno verso un aumento del peso totale della soluzione attuale, l'esecuzione di tale mossa abbandona le regole di un “approccio di discesa” e getta le basi per un processo di ricerca con tabù. La restrizione di fattibilità che richiede che sia prodotto un albero in ogni fase è significativa in questa illustrazione, dato che in generale le metodologie TS possono includere traiettorie di ricerca che violano i vari tipi di condizioni di fattibilità.

Fig. 13 Tipi di mosse di scambio



Dato un insieme di mosse, come lo scambio che ho selezionato per il mio esempio, il passo successivo è quello di scegliere gli attributi chiave che verranno utilizzati per la classificazione tabù. Tabu Search è molto flessibile in questa fase del progetto. Il problema di conoscenze specifiche può essere utilizzato come una guida per risolvere una struttura particolare. Nei problemi in cui sono definite le mosse con l'aggiunta ed eliminazione di elementi, le etichette di questi elementi possono essere utilizzate come attributi per far rispettare lo stato di tabù. Qui, in questo esempio, posso semplicemente fare riferimento ai rami come attributi della mossa, poiché la condizione di essere dentro o fuori l'albero, che è una proprietà che contraddistingue la soluzione corrente, può essere assunta sempre come automaticamente conosciuta dalla rappresentazione di una soluzione ragionevole.

La scelta di Classificazioni Tabu

Le classificazioni Tabu non devono essere simmetriche, cioè la struttura tabù può essere progettata per trattare aggiunte e cancellazioni degli elementi in modo diverso. Supponiamo per esempio che dopo la scelta dello scambio statico di Figura 13, che aggiunge il ramo(4,6) e toglie il ramo(4,7), sia stato assegnato un tabù per entrambi questi rami. Allora una possibilità

è quella di classificare questi due rami tabù-attivi per lo stesso numero di iterazioni. Ribadendo quanto già detto nelle sezioni precedenti, la disponibilità dei tabù ha significati diversi a seconda se il limite viene aggiunto od eliminato. Per un ramo aggiunto, tabù-attivo significa che a questo nodo non è permesso di essere eliminato dall'albero corrente per il numero di iterazioni che definisce il suo mandato tabù. Per un ramo eliminato, invece, tabù-attivo significa che col limite non è consentito inserirlo nella soluzione corrente durante la sua permanenza in carica tabù. Poiché ci sono molti più spigoli fuori l'albero che nella struttura, per l'attuazione di una struttura sembra ragionevole mantenere tabù un attributo da poco eliminato per un periodo di tempo più lungo di un attributo aggiunto di recente.

Si noti anche che per questo problema il tabù-attivo per il periodo di aggiunta dei rami è delimitato da k , dal momento che a nessun ramo aggiunto è permesso di essere eliminato per k iterazioni, quindi tutte le mosse disponibili saranno classificate tabù entro k passi. Il concetto di creazione asimmetrica nelle classificazioni tabù può essere facilmente applicata alle impostazioni dove mosse di aggiunta o eliminazione non vengono utilizzate.

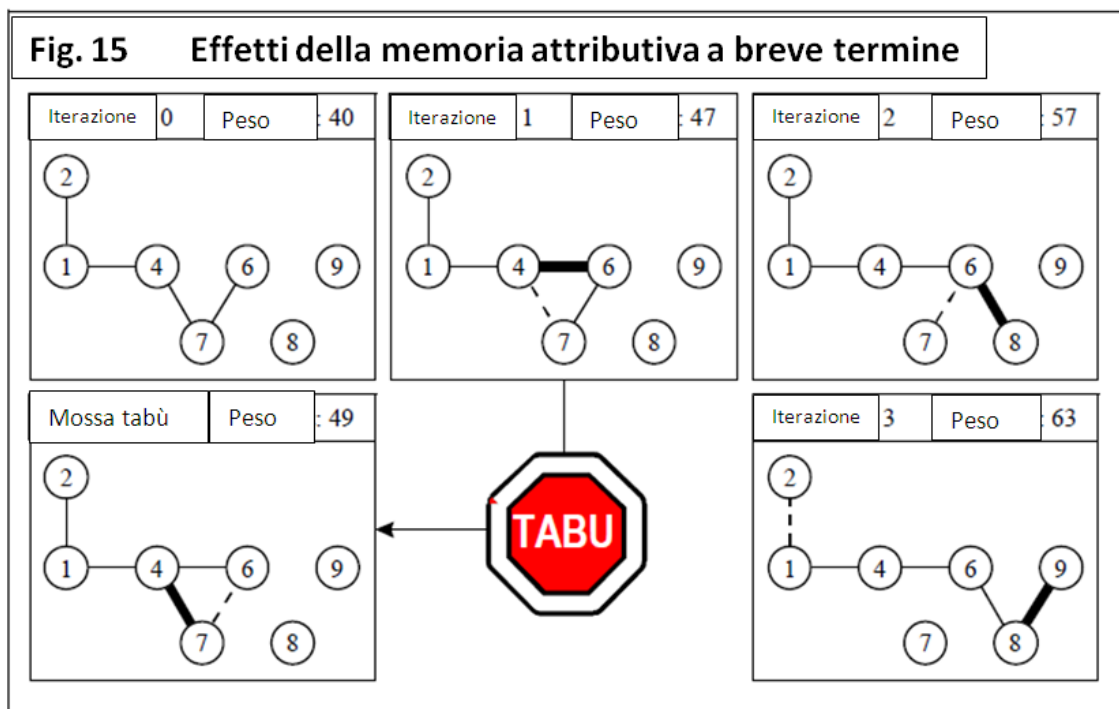
Illustrazione delle classificazioni tabù per il problema dell'albero-k minimizzato

Come precedentemente sottolineato, la classificazione tabù-attiva può infatti impedire la visita nella ricerca delle soluzioni che non sono state ancora esaminate. È di seguito illustrato questo fenomeno.

Supponiamo che nell'esempio del problema di minimizzazione dell'albero-k di Figura 11, i rami tolti siano tenuti tabù-attivi per 2 iterazioni, mentre i rami aggiunti siano tenuti tabù-attivi per una sola iterazione. Il numero di iterazioni è conservato in un margine tabù-attivo che è chiamato *tabu tenure* del ramo. Anche supporre che si definisca una mossa di scambio tabù sia se il suo ramo è aggiunto che se sia eliminato è tabù-attivo. Se esaminiamo l'intorno di rami disponibili per gli scambi ad ogni iterazione e scelgo sempre il migliore che non è tabù, le prime tre mosse sono come indicate nella Tabella 14 di seguito, a partire dalla soluzione iniziale trovata dall'euristico con costruzione golosa. La mossa di iterazione 1 è la mossa di scambio statica precedentemente identificata nella Figura 13. Schemi che mostrano gli alberi successivi generati da queste mosse, iniziando con la soluzione iniziale golosa, sono riportati nella Figura 15 (pagina seguente).

Iterazioni	Permanenza netta Tabu attiva		Aggiunta	Eliminazione	Peso
	1	2			
1			(4,6)	(4,7)	47
2	(4,6)	(4,7)	(6,8)	(6,7)	57
3	(6,8), (4,7)	(6,7)	(8,9)	(1,2)	63

I valori di permanenza netta di 1 e 2 nella Tabella 14 per i rami attualmente tabu-attivi indicano il numero di iterazioni che questi rami rimarranno tabu-attivi, compresa l'iterazione corrente.



All'iterazione 2, l'inversione del movimento di iterazione 1, vale a dire la mossa che aggiunge (4,7) ed elimina (4,6), è chiaramente tabù, dal momento che entrambi i suoi rami sono tabu-attivi a iterazione 2. Inoltre, la mossa che aggiunge (4,7) ed elimina (6,7) è anche classificata tabù perché contiene l'estremità tabù attiva (4,7), con un mandato netto di 2. Questa mossa porta ad una soluzione con un peso totale di 49, una soluzione che evidentemente non è stata visitata prima (vedi Figura 15). La classificazione tabu-attiva di (4,7) ha modificato l'intorno originale della soluzione di iterazione 2, ed ha costretto alla ricerca di scegliere una mossa con un valore peggiore di funzione obiettivo, ad esempio, quello con un peso totale di 57. In

questo caso, escludere la soluzione con un peso totale di 49 ha poco effetto sulla qualità della miglior soluzione trovata, dal momento che ne ho già ottenuta una con un peso di 40.

In altre situazioni, tuttavia, devono essere prese ulteriori precauzioni per evitare di perdere una buona soluzione. Queste strategie sono note come “criteri di aspirazione”. Forme alternative di criteri di aspirazione sono molto importanti nella ricerca con tabù. Per il momento osservo semplicemente che se la soluzione tabù incontrata nella fase attuale, invece aveva un peso di 39 che è migliore rispetto al peso di 40 fino ad ora visto, poi mi avrebbe permesso la classificazione tabù di questa soluzione che deve essere sovrascritta e consentire alla soluzione ammissibile di essere visitata. Il criterio di aspirazione che si applica in questo caso è chiamato criterio di “maggior aspirazione”. È importante tenere a mente che i criteri di aspirazione non costringono mosse particolari, da selezionare, ma semplicemente mettono a disposizione, o alternativamente recedono sanzioni di attributi collegati a certe classificazioni tabù.

Merita di essere fatto a questo punto un altro commento sulla classificazione tabù. Nella mia precedente discussione sul problema di minimizzare l'albero-k considero una mossa di scambio tabù se sia il suo ramo aggiunto che il ramo eliminato sono tabù-attivi. Tuttavia, si potrebbe invece stabilire che una mossa di scambio è tabù solo se entrambi, il suo ramo aggiunto e quello che elimino, sono tabù-attivi. In generale, lo stato tabù di una mossa è funzione degli attributi tabù-attivi delle mosse, cioè della nuova soluzione prodotta dal movimento.

Primo approccio

Ora ho a disposizione ingredienti sufficienti per un primo livello di procedura di ricerca con tabù. Tale procedura è a volte implementata in una prima fase di uno sviluppo TS per ottenere una prima idea delle caratteristiche di prestazione e calibrazione, o semplicemente per fornire un comodo approccio graduale allo scopo di eliminare gli errori del software di soluzione. Anche se questa forma semplificata con memoria a breve termine del metodo TS omette una serie di considerazioni importanti, e non include ancora le considerazioni a lungo termine, dà comunque un utile punto di partenza per dimostrare diversi aspetti fondamentali della ricerca con tabù.

Parto dalla soluzione con un peso di 63 come indicato in precedenza nella Figura 15, che è stata trovata all'iterazione 3. Ad ogni passo seleziono il peso minimo con le mosse non tabù ammissibili, ed uso il criterio di maggior aspirazione per permettere ad una mossa di essere considerata

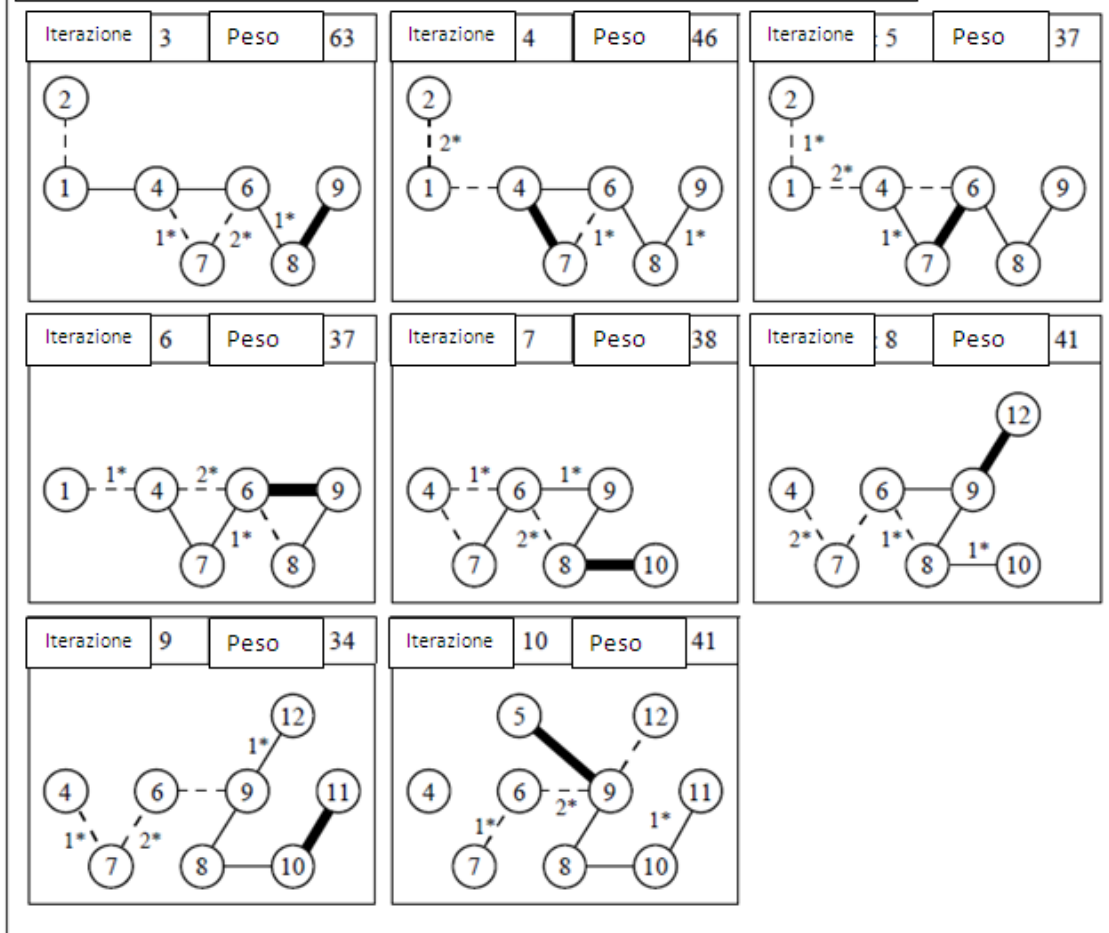
ammissibile nonostante conduca ad una soluzione tabù. Si può verificare che il risultato porta alla serie di soluzioni riportate nella Tabella 16, che continua dall'iterazione 3, appena eseguita.

Per semplicità, ho selezionato un dominio arbitrario di arresto che chiude la ricerca in 10 iterazioni.

Tab. 16 Iterazione ad un primo livello della procedura TS						
Iterazione	Possesso netto tabù-attivo		Aggiunta	Eliminaz.	Valore mossa	Peso
	1	2				
3	(6,8), (4,7)	(6,7)	(8,9)	(1,2)	6	63
4	(6,7), (8,9)	(1,2)	(4,7)	(1,4)	-17	46
5	(1,2), (4,7)	(1,4)	(6,7)	(4,6)	-9	37*
6	(1,4), (6,7)	(4,6)	(6,9)	(6,8)	0	37
7	(4,6), (6,9)	(6,8)	(8,10)	(4,7)	1	38
8	(6,8), (8,10)	(4,7)	(9,12)	(6,7)	3	41
9	(4,7), (9,12)	(6,7)	(10,11)	(6,9)	-7	34*
10	(6,7), (10,11)	(6,9)	(5,9)	(9,12)	7	41

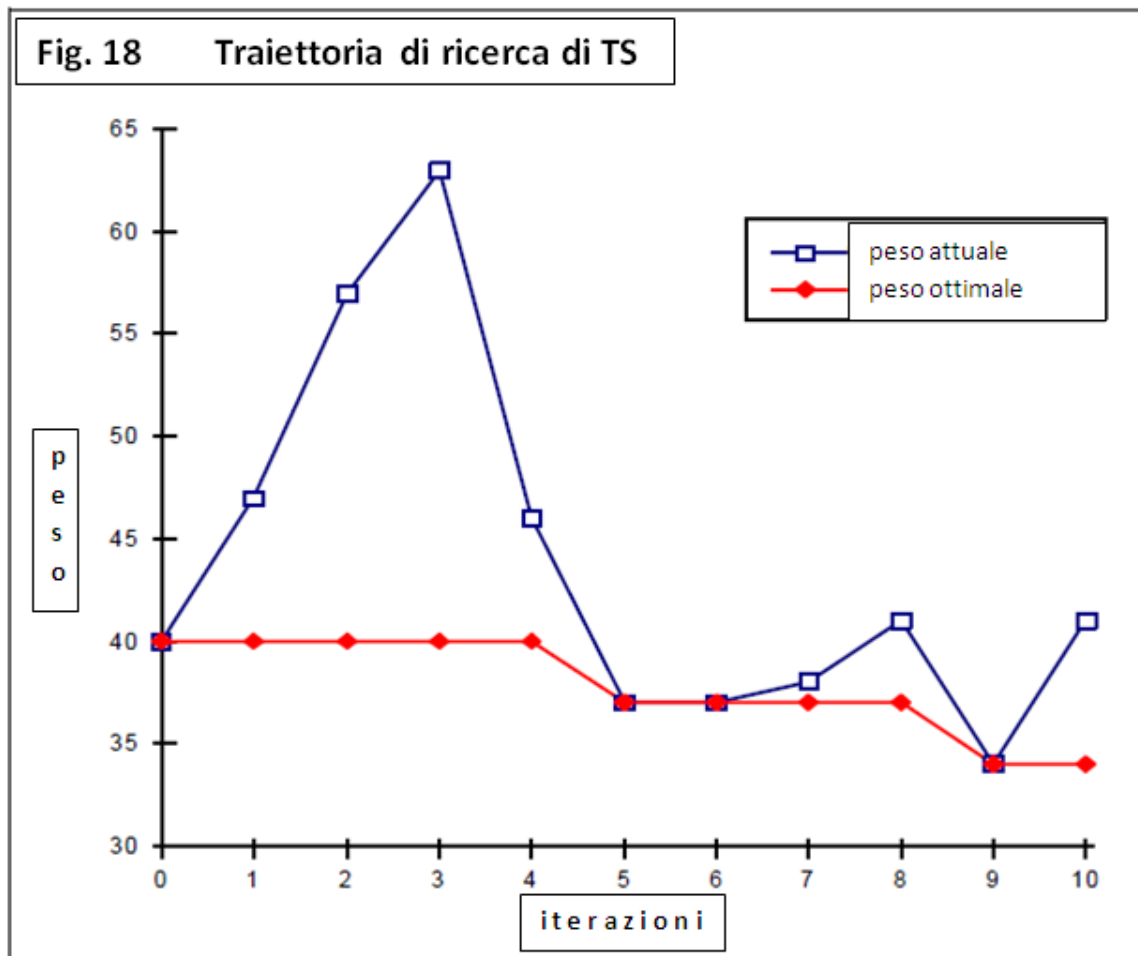
Le soluzioni successive identificate nella Tabella 16 sono riportate graficamente nella Figura 17 (pagina seguente). Oltre ad identificare il ramo che viene eliminato ad ogni passo con una linea tratteggiata, indico anche il ramo eliminato dal passo immediatamente precedente, con una linea tratteggiata che riporta 2*, ad indicare il suo mandato attuale tabù netto di 2. Allo stesso modo, identifico l'estremità eliminata un passo più indietro con una linea tratteggiata che riporta 1*, per indicare il suo attuale livello tabù netto di 1. Infine, anche il ramo che è stato aggiunto sul gradino immediatamente precedente è etichettato con 1* indicando che ha un possesso tabù attuale netto di 1. Così i rami che sono etichettati con possessi tabù sono quelli che attualmente sono tabù-attivi e che sono esclusi dall'essere scelti da una mossa nell'iterazione corrente, a meno del permesso di essere scelti grazie al criterio di aspirazione. Come illustrato nella Tabella 16 e Figura 17, il metodo continua a generare diverse soluzioni e nel tempo la migliore soluzione nota, contrassegnata da un asterisco, migliora progressivamente. Infatti, può essere verificato per questo semplice esempio che la soluzione ottenuta all'iterazione 9 è ottimale. In generale, ovviamente, non vi è alcun modo conosciuto per verificare l'ottimalità di difficili problemi di ottimizzazione discreta, cioè quelli che rientrano nella classe chiamata *NP-hard*. Il problema di minimizzazione di un albero-k è uno di questi.

Fig. 17 Rappresentazione grafica delle iterazioni di TS



Si può osservare che all'iterazione 6 il metodo ha scelto una mossa con un valore di spostamento pari a zero. Tuttavia, la configurazione dei cambiamenti della soluzione corrente dopo l'esecuzione di queste mosse è illustrata in Figura 17.

La selezione delle mosse con alcuni valori di spostamento, come ad esempio spostare i valori a zero, può essere strategicamente controllata, col limitare con particolari impostazioni la loro scelta come assicurazione aggiuntiva contro il cicliismo. Mostrerò presto come, al di là di considerazioni su questa implementazione di primo livello, può portare a cercare una traiettoria migliore in graduale miglioramento ed il comportamento è in generale caratteristico di TS. La Figura 18 (pagina seguente) fornisce una rappresentazione grafica di questo comportamento per l'esempio attuale.



Ho volutamente scelto che l'iterazione si fermasse ad essere piccola per illustrare un'ulteriore rilevante funzionalità e dare un elemento per considerare alcuni tipi di valutazioni a più lungo termine.

Un modo naturale per applicare TS è di interrompere periodicamente il suo progresso, soprattutto se il suo tasso nel trovare nuove soluzioni migliori scende sotto un livello predefinito, e per riavviare il metodo da una fase designato per generare una nuova sequenza di soluzioni. Il classico riavvio delle procedure sulla base di "randomizzazione" evidentemente può essere utilizzato per questo scopo, ma TS trae spesso un vantaggio impiegando forme più strategiche di riavvio. Illustro un semplice esempio di tale procedura di riavvio, che serve anche per introdurre un concetto utile di memoria.

La memoria di eventi critici

La memoria degli eventi critici in TS, come suggerisce il nome, controlla il verificarsi di alcuni eventi critici durante la ricerca, e stabilisce una memoria che costituisce un aggregato sintesi di

questi eventi. Per il mio esempio, in cui cerco di generare una nuova soluzione di partenza, un evento critico che è chiaramente rilevante è la generazione della partenza da una precedente soluzione. Di conseguenza, se si applica una procedura di riavvio più volte, le fasi di generazione di tutte le soluzioni precedenti di partenza naturalmente si qualificano come eventi critici. Cioè, ho preferito discostarmi da tali soluzioni in qualche maniera significativa come ho generato altre soluzioni di partenza.

Diversi livelli di partenza, che rappresentano diversi livelli di diversificazione, possono essere raggiunti attraverso la definizione di soluzioni che corrispondono ad eventi critici in modi diversi, ed attivando con la memoria di eventi critici delle regole diverse. Nel contesto attuale ritengo importante che nuove soluzioni di partenza non solo differiscano dalle precedenti soluzioni di partenza, ma che differiscano anche da altre soluzioni generate durante passaggi precedenti. Una possibilità è quella di utilizzare un approccio coperto che considera ogni soluzione completa generata in precedenza per rappresentare un evento critico. L'aggregazione di tali eventi attraverso la memoria dell'evento critico lo rende del tutto possibile, ma spesso è più che sufficiente, e a volte preferibile, per isolare un insieme ridotto di soluzioni.

Per l'esempio attuale, pertanto, si precisano che gli eventi critici di interesse costituiti da generazione non solo della soluzione di partenza del passo precedente, ma anche ogni successiva soluzione che rappresenta un "ottimo locale TS", cioè il cui obiettivo è una migliore funzione di valore (o no) peggiore di quella della soluzione immediatamente prima e dopo di esso. Utilizzando questa semplice definizione vedo che quattro soluzioni si qualificano come critiche, cioè vengono generate dagli eventi indicati come critici nel passaggio di prima soluzione del mio esempio: la soluzione iniziale e le soluzioni trovate alle iterazioni 5, 6 e 9 con pesi rispettivamente di 40, 37, 37 e 34. Dal momento che la soluzione all'iterazione 9 sembra essere ottimale, mi interessa che sia trovato l'effetto di riavvio prima di questa soluzione. Suppongo di aver scelto di riavviare dopo l'iterazione 7, senza tuttavia raggiungere una soluzione ottimale. Allora le soluzioni che corrispondono a eventi critici sono la soluzione iniziale e le soluzioni di iterazioni 5 e 6. Tratto queste tre soluzioni in forma aggregata, combinando i loro rami, per creare un sottografo che consiste dei rami (1,2), (1,4), (4,7), (6,7), (6,8), (8,9) e (6,9). La memoria basata sulla frequenza affina questa rappresentazione dal punto di vista contabile per il numero di volte che ogni nodo compare nella soluzione di importanza critica, e consente l'inclusione dei fattori di ponderazione aggiuntivi. Per eseguire una procedura di riavvio, mi penalizza l'inserimento dei rami di questo sottografo a varie fasi di costruzione della nuova soluzione. Di solito è preferibile applicare questo processo di penalizzazione ai primi passi, consentendo implicitamente alla funzione di penalizzazione di

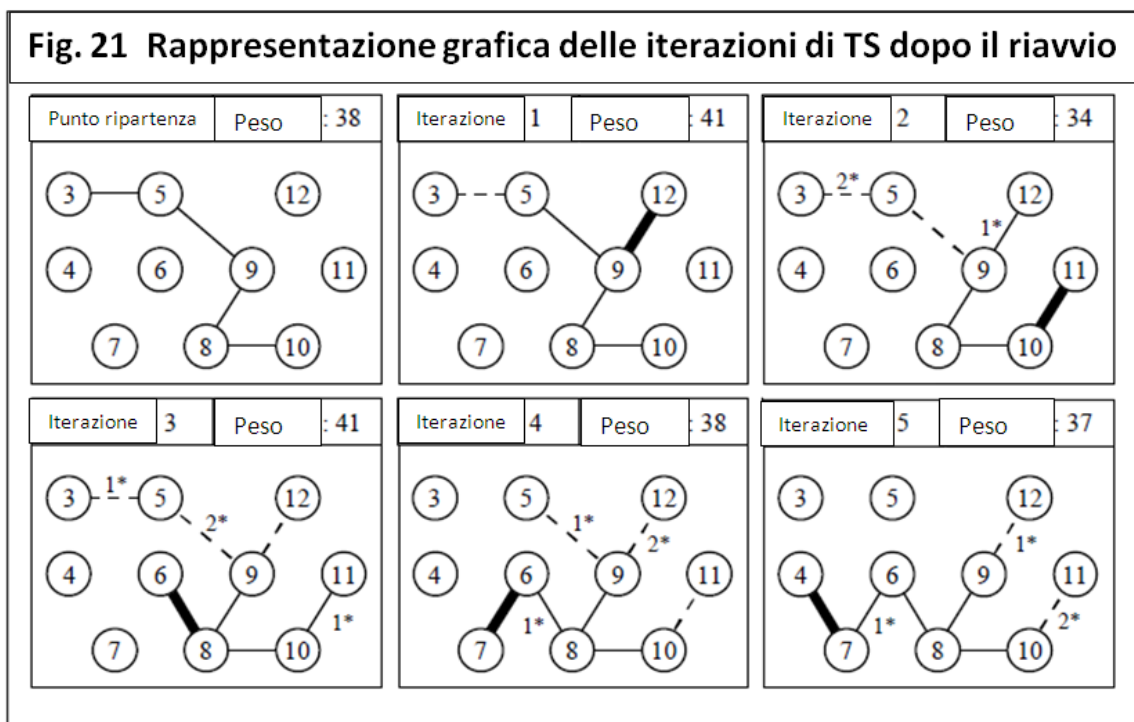
decadere rapidamente con il numero di incrementi di passi. A volte è anche utile per consentire uno o più passi dopo l'intervento di applicazione di tali sanzioni, prima di applicarle nuovamente. Per questo esempio, userò la memoria inclusa nel sottografo di rami penalizzati dall'introduzione di un peso di grandi dimensioni che di fatto esclude tutti questi rami dalla considerazione nella prima di due fasi di costruzione della nuova soluzione. Poi, perché la costruzione si articola in quattro passi in totale, non sarà attivato l'evento di memoria critica nelle successive fasi di costruzione, ma permetterà al metodo di procedere nella sua forma iniziale. L'applicazione di questo approccio, riavvia il metodo con il primo ramo scegliendo (3,5), che è il peso minimo del ramo non nel sottografo penalizzato. Questa scelta e le scelte rimanenti che generano la nuova soluzione di partenza sono riportate nella Tabella 19.

Tab. 19 Procedura di riavvio			
Passo	Candidati	Selezione	Peso totale
1	(3,5)	(3, 5)	6
2	(2,3), (3,4), (3,6), (5,6), (5,9), (5,12)	(5, 9)	22
3	(2,3), (3,4), (3,6), (5,6), (5,12), (6,9), (8,9), (9,12)	(8, 9)	29
4	(2,3), (3,4), (3,6), (5,6), (5,12), (6,8), (6,9), (7,8), (8,10), (9,12)	(8, 10)	38

A partire dalla soluzione costruita nella Tabella 19, e applicando la prima procedura di TS a livello esattamente come è stato applicato al primo passaggio, si genera la sequenza delle soluzioni riportate nella Tabella 20 e rappresentate in Figura 21 (pagina seguente). Anche in questo caso, ho arbitrariamente limitato il numero totale di iterazioni, in questo caso a 5.

Tab. 20 Iterazioni di TS per il riavvio successivo						
Iterazione	Possesso tabu-attivo netto		Aggiunta	Eliminaz.	Valore mossa	Peso
	1	2				
1			(9,12)	(3,5)	3	41
2	(9,12)	(3,5)	(10,11)	(5,9)	-7	34*
3	(3,5), (10,11)	(5,9)	(6,8)	(9,12)	7	41
4	(5,9), (6,8)	(9,12)	(6,7)	(10,11)	-3	38
5	(9,12), (6,7)	(10,11)	(4,7)	(8,10)	-1	37

E' interessante notare che la procedura genera il riavvio di una soluzione migliore, per un peso totale di 38, rispetto alla soluzione iniziale generata durante la prima costruzione, per un peso totale di 40. Inoltre, la soluzione di riavvio contiene 2 "rami ottimali" (cioè, i rami che sembrano appartenere alla struttura ottimale). Questa soluzione permette di iniziare la traiettoria di ricerca per trovare la soluzione ottimale in sole due iterazioni, illustrando i vantaggi dell'applicazione di un evento di memoria critica all'interno di una strategia di ripresa. Strutture di memoria relative possono anche essere preziose per le strategie che guidano la ricerca in nuove regioni dal "riavvio parziale" o direttamente da una traiettoria continua corrente, con regole decisionali modificate.



Nella sua forma completa, Tabu Search contiene una varietà di elementi che vanno oltre questo primo livello di preoccupazioni e che aprono possibilità per la creazione di approcci più potenti alla soluzione. Una relativa collezione di principi di soluzioni per problemi sta emergendo ed invita ad una più completa esplorazione come base per sfruttare la struttura in contesti diversi.

8 CONCLUSIONE

Con questa tesi ho dimostrato come Tabu Search sia un potente approccio algoritmico che è stato applicato con grande successo a molti difficili problemi combinatori. Una caratteristica particolarmente positiva di TS è che, come tutti gli approcci basati sulla ricerca locale, può facilmente gestire vincoli "grossolani" di complicazioni che si trovano tipicamente in applicazioni reali. È quindi un approccio molto pratico. Non è tuttavia una panacea: ogni revisore o editore di una rivista scientifica si è dovuto scontrare in diverse misure con il fallimento euristico di TS. Questi fallimenti derivano da due cause principali: un'insufficiente comprensione dei concetti fondamentali del metodo, ma anche, più spesso, la mancanza di comprensione completa del problema sotto mano. Non si può sviluppare un buon TS euristico per un problema che non sia ben conosciuto. Questo perché una conoscenza significativa del problema è assolutamente necessaria per eseguire le operazioni più basilari dello sviluppo di qualsiasi procedura di TS, e cioè la scelta di uno spazio di ricerca e di una struttura di intorni efficaci. Se lo spazio di ricerca e la struttura di intorni non sono sufficienti, nessuna competenza di TS sarà sufficiente a salvare la situazione. Per avere successo, tutti i meta-euristici necessitano sia di profondità che di larghezza per ottenere risultati nel loro processo di ricerca, profondità di solito non è un problema per TS, che è abbastanza aggressivo in questo senso (l'euristico TS nella ricerca trova molto presto soluzioni generalmente abbastanza buone), ma la larghezza può essere un problema critico. Per gestire questa situazione, è estremamente importante sviluppare un sistema efficace di diversificazione.

I successi pratici di tabu search hanno promosso la ricerca utile in modo da sfruttare la base delle sue idee nel modo più completo. Allo stesso tempo, molti aspetti di queste idee devono ancora essere esplorati. Le questioni di individuare le migliori combinazioni di memoria a breve e a lungo termine e di migliori equilibri di strategie di intensificazione e diversificazione contengono ancora molte aree non esaminate, e alcune di queste indubbiamente porteranno a scoperte importanti per lo sviluppo di metodi di soluzione più potenti per il futuro.

Ci sono evidenti contrasti tra punti di vista ma l'opinione di TS è attualmente favorita rispetto all'intelligenza artificiale e alle reti neurali, in particolare riguardo al ruolo della memoria nella ricerca. Tuttavia, ci sono anche utili complementarità tra questi punti di vista, che sollevano la possibilità di realizzare sistemi che integrano le loro qualità fondamentali. I progressi sono già in corso in questo campo, con la creazione di modelli di apprendimento e formazione Tabù (de Werra e Hertz (1989), Beyer e Orgier (1991), Battiti e Tecchioli (1993), Gee e Prager (1994)), le macchine tabu (Chakrapani e Skorin-Kapov (1993), Nemati e Sun (1994)) e le procedure di

progettazione Tabu (Kelly e Gordon (1994)). Negli ultimi anni si è assistito indubbiamente a significativi aumenti della difficoltà di soluzione dei problemi di ottimizzazione, ma si deve anche riconoscere che resta molto da imparare. La ricerca in queste aree è piena di settori inesplorati ed interessanti.

Altre procedure e concetti collegati all'algoritmo Tabu Search, ma che non ho considerato in questa tesi, sono l'influenza, la misura di distanze e diversità, mosse composte, profondità variabile, catena di espulsione, il principio di prossima ottimalità, il principio di strutture congeniali, il principio della piramide, Tabu Search probabilistico, usi del processo in parallelo, il principio spazio/tempo, l'analisi-obiettivo, la costruzione di lessicari e diversi altri elementi.

9 BIBLIOGRAFIA

Battiti, R. and G. Tecchiolli (1994) "The Reactive Tabu Search", *ORSA Journal on Computing*, Vol. 6, No. 2, A description of a specialized tabu search implementation that seeks to dynamically control search parameters with feedback mechanisms.

Costa D. -A Tabu Search Algorithm for Computing an Operational Time Table - *European Journal of Operational Research* 76, (1994), pp. 98-110

Crainic, T. G., M. Toulouse and M. Gendreau (1997) "Toward a Taxonomy of Parallel Tabu Search Heuristics," *INFORMS Journal on Computing*, Vol. 9, No. 1, pp. 61-72. A broad view of alternative designs for parallel tabu search implementations.

Dell'Amico, M. and M. Trubian (1993) "Applying Tabu Search to the Job-Shop Scheduling Problem," *Annals of Operations Research*, Vol. 41, 231-252. A good example of a very effective implementation of tabu search in the context of production scheduling.

Faigle, U. and W. Kern (1992) "Some Convergence Results for Probabilistic Tabu Search," *ORSA Journal on Computing*, Vol. 4, No. 1, pp. 32-38. With some simplifying assumptions, the authors of this article present convergence results for the probabilistic variant of tabu search.

Friden C. , Hertz A. , de Werra D. -*STABULUS: a technique for finding stable sets in large graphs with tabu search* - *Computing* 42, (1989) pp. 35-44.

Gendreau M., Hertz A., Laporte G. -A Tabu Search Heuristic for the Vehicle Routing Problem- *Management Science* 40/10, (1994), pp. 1276-1290.

Glover, F. (1977) "Heuristics for Integer Programming Using Surrogate Constraints," *Decision Sciences*, Vol 8, No 1, 156-166. Seminal work on tabu search and scatter search.

Glover, F. (1989) "Tabu Search — Part I," *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190-206. First comprehensive description of tabu search.

Hansen P. -*The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming* - Presented at the Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy, (1986)

Hertz A. -*Tabu Search for Large Scale Timetabling Problems* - European Journal of Operational Research 54/1, (1991), pp. 39-47.

Lin S. , Kernighan B.W. -*An Effective Heuristic Algorithm for the Traveling-Salesman Problem* - Operations Research 21, (1973), pp. 498-516.

Osman I.H. -*Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem* - Annals of Operations Research. 41, (1993), pp. 421-451

Rego, C. (1998) "A Subpath Ejection Method for the Vehicle Routing Problem," *Management Science*, Vol. 44, No. 10, pp. 1447-1459. An example of a tabu search implementation that employs ejection chain mechanisms to define effective solution neighborhoods.

Semet F., Taillard E. -*Solving real-life vehicle routing problems efficiently using tabu search* - Annals of Operations Research 41, (1993), pp. 469-488.

Taillard E. -*Robust Taboo Search for the Quadratic Assignment Problem* - Parallel Computing 17, (1991), pp. 443-455.

de Werra D., Hertz A. -*Tabu Search Techniques: A tutorial and an application to neural networks* - OR Spektrum, (1989), pp. 131-141.

Ringraziamenti

Un sincero ringraziamento va *in primis* al Prof. Giorgio Romanin Jacur per avermi consigliato il tema di questo lavoro, che ho poi sviluppato con piacere ed interesse. Lo ringrazio in particolare per la disponibilità, la cortesia e le attenzioni nel darmi importanti consigli che ho fedelmente seguito.

Ringrazio molto la mia splendida famiglia, Elio, Brunella, Andrea, Elda e zii e cugini vicini e lontani per avermi dato la possibilità di studiare nel modo più sereno e motivato possibile.

Ringrazio la mia ragazza speciale, Valentina, che mi ha sopportato nello sforzo di raggiungere questo traguardo e sempre mi sosterrà nel prosieguo dei miei studi e nella carriera lavorativa.

Ringrazio i miei amici più cari, sparsi per l'Italia, il Gemello, gli Andrea, i Giacomi, gli Stefani, Alberto, Marcello, mia sorella acquisita Carlotta, la mia squadra di calcio del Limana, i miei compagni di studi ed i miei coinquilini dell'affiatatissima Reggia, Simone e Piero, per avermi insegnato ognuno qualcosa.

Grazie infine a tutti i Professori e a tutte le persone che nel corso di questi anni hanno contribuito alla formazione della mia persona, non solo dal punto di vista professionale.