

Università degli Studi di Salerno



Facoltà di Scienze Matematiche Fisiche e Naturali

Dipartimento di Matematica

Dottorato di Ricerca in Matematica

XII Ciclo – Nuova Serie

TESI DI DOTTORATO

**Ottimizzazione multi-obiettivo:
scheduling delle attività di
manutenzione delle reti
elettriche.**

CANDIDATA: **DOTT.SSA VINCENZA TUFANO**

COORDINATORE: **PROF.SSA PATRIZIA LONGOBARDI**

TUTOR: **PROF. ABDELAZIZ RHANDI**

CO-TUTOR: **DOTT.SSA ROSANNA MANZO**

Anno Accademico 2012 – 2013

Indice

Introduzione	i
1 Problemi di scheduling	1
1.1 Definizione dei problemi di scheduling	3
1.2 La schedulazione del personale in letteratura	16
2 Schedulazione degli interventi di manutenzione delle reti elettriche	33
2.1 Informazioni di ingresso	36
2.2 Formulazione matematica del problema di scheduling	37
3 Tecniche di schedulazione degli interventi di manutenzione	48
3.1 Tecniche basate su Evolutionary Computation (EC)	49
3.2 Applicazione delle tecniche sul caso di studio	51
3.2.1 Confronto tra tecniche di risoluzione e risultati ottenuti applicando il modello matematico	52
Conclusioni	68

Elenco delle figure

1.1	Problema di rostering in fasi	5
1.2	Esempio di mossa	12
1.3	Esempio di pattern di lavoro	18
1.4	Procedura pilota-pilota assegnazione iniziale	29
1.5	Procedura “pilota per pilota” - assegnazione dopo la prima mossa	30
3.1	Confronto dei tempi di esecuzione delle metauristiche <i>NSGA</i> e <i>NS-GAII</i> con il tempo ottenuto dal software <i>CPLEX</i>	62
3.2	Confronto tecniche sull’ <i>Istanza A</i> in base al parametro <i>Costo di Spostamento</i>	62
3.3	Confronto tecniche sull’ <i>Istanza B</i> in base al parametro <i>Costo di Spostamento</i>	63
3.4	Confronto tecniche sull’ <i>Istanza C</i> in base al parametro <i>Costo di Spostamento</i>	63
3.5	Confronto tecniche sull’ <i>Istanza A</i> in base al parametro <i>Costo di Straordinario</i>	64
3.6	Confronto tecniche sull’ <i>Istanza B</i> in base al parametro <i>Costo di Straordinario</i>	64

3.7	Confronto tecniche sull' <i>Istanza C</i> in base al parametro <i>Costo di Straordinario</i>	65
3.8	Confronto tecniche sull' <i>Istanza A</i> in base al parametro <i>Tempo di interruzione</i>	65
3.9	Confronto tecniche sull' <i>Istanza B</i> in base al parametro <i>Tempo di interruzione</i>	66
3.10	Confronto tecniche sull' <i>Istanza C</i> in base al parametro <i>Tempo di interruzione</i>	66

Elenco delle tabelle

3.1	Risultati ottenuti applicando la tecnica <i>NSGA</i> sull' Istanza A	53
3.2	Risultati ottenuti applicando la tecnica <i>NSGA</i> sull' Istanza B	54
3.3	Risultati ottenuti applicando la tecnica <i>NSGA</i> sull' Istanza C	55
3.4	Risultati ottenuti applicando la tecnica <i>NSGA II</i> sull' Istanza A	56
3.5	Risultati ottenuti applicando la tecnica <i>NSGA II</i> sull' Istanza B	57
3.6	Risultati ottenuti applicando la tecnica <i>NSGA II</i> sull' Istanza C	58
3.7	Risultati ottenuti utilizzando il software <i>CPLEX</i> per risolvere il modello matematico sull' Istanza A	59
3.8	Risultati ottenuti utilizzando il software <i>CPLEX</i> per risolvere il modello matematico sull' Istanza B	60
3.9	Risultati ottenuti utilizzando il software <i>CPLEX</i> per risolvere il modello matematico sull' Istanza C	61

Introduzione

La disponibilità di energia elettrica è un fattore cruciale delle economie industrializzate di oggi essendo alla base di qualsiasi settore lavorativo. Al fine di garantire energia elettrica fino ai luoghi più remoti del territorio, fitte reti di linee e cabine elettriche sono dislocate lungo il territorio nazionale. Risulta chiaro che un parametro chiave da tenere sotto controllo è l'affidabilità, in genere valutata mediante l'indice SAIDI (System Average Interruption Index), che misura la durata media di interruzione per ogni cliente in un anno. Al fine di poter mantenere basso tale valore e garantire una elevata capillarità delle linee, è necessario prevenire e/o intervenire in tempi brevi su un guasto. A tal proposito nasce l'esigenza di elaborare politiche di intervento sulle linee elettriche, che mirino alla corretta gestione degli interventi ovvero alla corretta pianificazione degli interventi di manutenzione delle reti elettriche.

Attualmente molti gestori delle reti elettriche pianificano in modo manuale gli interventi di manutenzione ovvero le pianificazioni vengono elaborate giorno per giorno, ed un supervisore decide chi lavorerà su quale intervento. Il lavoro di quest'ultimo è arduo, in quanto, durante la fase di assegnazione vi è la necessità di tenere sotto controllo una moltitudine di parametri che sono difficili da integrare in un sistema di modellazione. Tali parametri possono essere ad esempio informazioni relative alla

disponibilità (se una risorsa è impegnata ad esempio in altri lavori e quindi non disponibile), alla locazione, che rappresenta la sede dalla quale la risorsa operativa comincia il suo turno di lavoro, o anche informazioni relative al tempo che la risorsa operativa impiega per raggiungere dalla sua posizione di partenza il luogo in cui sarà svolto il successivo task, etc. La modellazione di tale problema si colloca in una classe di problemi molto ampia, ovvero problemi di ottimizzazione multi-obiettivo e coinvolge simultaneamente questioni afferenti a tre classi di problemi:

- *Personnel Scheduling Problem*: associazione di turni o working days alla forza lavoro (rostering);
- *Task Assignment Problem*: associazione “ottima” di n agenti su m task, in modo tale che ad ogni task sia assegnato esattamente un agente;
- *Vehicle Routing Problem*: ottimizzazione degli itinerari, dato un insieme di veicoli ed un insieme di locazioni, geograficamente distribuite.

L’obiettivo di questo progetto di tesi è di studiare le tecniche di ottimizzazione multi-obiettivo applicate al problema dello scheduling delle attività di manutenzione delle reti elettriche, elaborare e progettare un algoritmo che sarà alla base di un tool più ampio che supporterà il supervisore nel suo dovere. Tale problema può essere descritto come segue. Dato un insieme di task di manutenzione che devono essere effettuati da un determinato gruppo di lavoratori o squadre bisogna trovare una giusta allocazione delle risorse ai task e una pianificazione degli interventi in un determinato arco temporale, che tenga in considerazione vari vincoli, alcuni in materia di lavoro del personale ovvero numero di ore lavorate in una giornata lavorativa, straordinari, etc., altri sulle competenze che devono essere possedute dalle risorse per risolvere un determinato intervento, e ancora vincoli sulle priorità dell’intervento. Pertanto, la

principale sfida proposta prevede lo sviluppo di un algoritmo di allocazione ottima delle risorse per risolvere il problema di massimizzare il numero degli interventi della forza lavoro rispettando i vincoli (quali skill o materiali necessari, appuntamenti o priorità) e garantendo allo stesso tempo buoni livelli di servizio e minimizzando i costi come kilometraggio e straordinario.

Il problema di ottimizzazione multi-obiettivo affrontato trova applicazioni in diversi ambiti operativi quali ad esempio strutture ospedaliere e assistenziali, aeroporti, enti militari.

La tesi è organizzata nel seguente modo. Nel Capitolo 1 viene condotto uno studio sugli algoritmi di schedulazione presenti in letteratura. Nella prima parte del capitolo vengono analizzati e studiati i problemi di scheduling ed i metodi e le tecniche per la loro risoluzione. In particolare, vengono analizzate le tecniche *Hill Climbing*, *Steepest Descent* e *Tabu Search* utilizzate, in combinazione con la programmazione con vincoli, per risolvere il problema di rostering in uno scenario ospedaliero. Nella seconda parte del capitolo vengono presentati due risoluzioni di problemi di scheduling, una soluzione a singolo obiettivo e una soluzione per problemi di tipo multi-obiettivo che applica la tecnica del *Simulated Annealing* per risolvere i cosiddetti problemi di Airlines Crew Rostering.

Nel Capitolo 2 viene formalizzato il modello matematico per risolvere il problema della schedulazione degli interventi di manutenzione elettrica.

Infine nel Capitolo 3 vengono applicate le tecniche di risoluzione dell'algoritmo su un insieme di dati forniti dal maggiore distributore di energia elettrica in Italia.

Capitolo 1

Problemi di scheduling

In molti settori lavorativi è necessario garantire una pronta risposta alle necessità del cliente e l'espletamento del servizio senza ritardi, e quindi spesso non secondo un orario regolare e costante nel tempo ma con un andamento temporale variabile a seconda della richiesta del cliente. Si pensi ad ospedali, catene di montaggio, call center, aziende di manutenzione elettrica ecc., che devono garantire elevati livelli di servizio, coprendo l'intero arco delle 24 ore. Ciò comporta una organizzazione e turnazione del personale al fine di assicurare la suddetta copertura, ovviamente rispettando i vincoli contrattuali e le esigenze del personale a disposizione. In piccole realtà lavorative, di solito, a vigilare sull'organizzazione del personale ci sono pochi addetti, che nella maggior parte dei casi operano senza alcun supporto tecnologico, o in modo limitato. Ma in realtà come quelle ospedaliere, o di aziende che si occupano della manutenzione della rete elettrica, in cui il numero di dipendenti soggetto a turnazione è elevato a causa dell'elevata richiesta della clientela, è doveroso per gli addetti agli uffici del personale dotarsi di sistemi atti a organizzare in modo ottimale il lavoro, garantendo al contempo la qualità del servizio e cercando di minimizzare

quanto possibile i relativi costi.

In questo lavoro di tesi il focus è incentrato sulla gestione/manutenzione delle reti elettriche, ed in particolare sull'ottimizzazione della qualità del servizio che caratterizza la necessità del cliente. Tale ottimizzazione è ottenibile esclusivamente rendendo sempre più veloce, versatile e flessibile la risposta della forza lavoro presente sul territorio. In letteratura il suddetto problema rientra nella categoria del Field Force Automation, ed è rivolto all'ottimizzazione dei processi e delle informazioni delle aziende che necessitano di mandare "sul campo" il proprio staff, coordinando simultaneamente gli interventi di più uomini in più luoghi, cercando di mantenere bassi i costi ed al contempo garantire un'elevata qualità del servizio erogato dal cliente.

Un lavoro di manutenzione può essere costituito da diversi task interdipendenti localizzati in differenti punti geografici. Quindi, nella pianificazione della forza lavoro, oltre ad assegnare i task alle squadre che dovranno effettuare l'intervento, bisogna valutare anche la distanza geografica tra la base dove è assegnata la squadra e la posizione dell'intervento, al fine di individuare una schedulazione ottimale delle squadre.

Il corrispondente problema di ottimizzazione che viene affrontato in questo lavoro di tesi può essere descritto nel modo seguente: dato un insieme di job di manutenzione che devono essere assolti da un insieme di squadre, il problema è individuare un assegnamento dei task alle squadre con l'obiettivo di:

- massimizzare la produttività;
- minimizzare i costi operativi;
- massimizzare la qualità del servizio.

1.1 Definizione dei problemi di scheduling

La schedulazione del personale, o turni del personale, può essere definita come il processo mirato alla definizione dei turni di lavoro del personale in modo da soddisfare le richieste di intervento previste in un determinato periodo temporale.

La prima parte di questo processo mira a definire il numero di persone, con particolari competenze, necessarie a soddisfare una determinata richiesta di intervento. Singole risorse, o membri della squadra, devono essere assegnati a turni in modo da soddisfare il livello di competenza richiesto alla squadra per risolvere un intervento e gli interventi sono poi assegnati agli individui per ogni turno. Durante il processo di scheduling devono essere considerati tutti i regolamenti industriali associati ai contratti di lavoro.

E' estremamente difficile trovare delle buone soluzioni a questi problemi altamente vincolati e complessi, ed è anche difficile determinare le soluzioni ottime che minimizzano i costi, rispondono alle esigenze dei dipendenti, distribuiscono equamente i turni tra i dipendenti e soddisfano tutti i vincoli sul posto di lavoro.

Nasce, quindi, l'esigenza dello sviluppo di modelli matematici e algoritmi che siano utili alla schedulazione dei turni. In particolare tale sviluppo comporterà:

- a) uno studio di modellazione delle richieste, che raccoglie e utilizza i dati storici per prevedere le richieste di servizio e convertirle nei livelli di competenza della squadra per soddisfare gli standard della richiesta;
- b) considerazioni sulle tecniche di soluzione necessarie per la creazione di uno strumento di schedulazione del personale che soddisfi i vincoli derivanti dalla normativa sul lavoro, sebbene meglio soddisfi una serie di obiettivi tra cui la copertura del fabbisogno del personale, costi minimi e la massima soddisfazione dei dipendenti;

c) la specificazione di uno strumento di reporting che consenta di visualizzare le soluzioni e fornisca report di prestazioni.

Il processo di turnazione del personale, chiamato anche *rostering* o *staff scheduling*, è comune a molte organizzazioni e può essere applicato quindi a un gran numero di settori come ad esempio:

- Trasporti;
- Ospedali;
- Call center;
- Equipaggio compagnie aeree.

Si deve notare che sono molte le attività lavorative in cui il servizio non può essere svolto secondo un orario regolare e costante nel tempo, ma segue un andamento temporale variabile a seconda della domanda esterna, che per le sue caratteristiche, richiede che il servizio venga soddisfatto senza ritardi.

La variabilità temporale dipende dal fatto che il servizio riguarda ad esempio il trasporto e quindi gli orari di servizio sono quelli legati al trasporto.

In questi casi, dunque, bisogna stabilire dei turni variabili. Si tratta di problemi complessi perchè si deve cercare un giusto equilibrio tra livello di servizio verso l'utenza e turni lavorativi compatibili con i contratti di lavoro e le normative di sicurezza.

Il processo di turnazione del personale viene solitamente suddiviso in fasi, durante le quali possono essere necessari più modelli per ogni specifico campo di applicazione.

In letteratura sono state proposte diverse classificazioni, in base al settore di applicazione e alla specificità del problema.

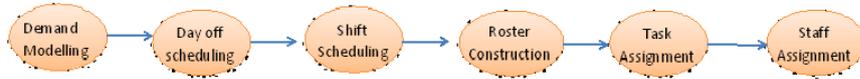


Figura 1.1: Problema di rostering in fasi

Lo sviluppo di un *roster* consiste nella determinazione del fabbisogno di personale con determinate competenze da impiegare durante il periodo di pianificazione, al fine di garantire un adeguato livello di servizio.

In [8] il processo di *rostering* viene definito, in modo generale, come un numero di modelli che parte dalla determinazione delle richieste di personale e termina con la specifica del lavoro (Fig. 1.1) che deve essere eseguito durante un periodo temporale da ogni individuo della squadra. Sebbene le fasi portino alla definizione di procedure step by step, lo sviluppo di un particolare *roster* richiede solo alcune fasi e, in particolari implementazioni, diverse fasi possono essere combinate in una procedura.

La prima fase nello sviluppo di un roster è detta *demand modelling* e consiste nella determinazione del numero di persone necessarie a soddisfare una richiesta in tempi diversi durante l'intervallo di pianificazione, o anche chiamato *rostering horizon*. Per poter determinare il fabbisogno del personale, qualora non sia conosciuto a priori, è necessario utilizzare delle tecniche al fine di prevedere il verificarsi di alcuni eventi, tecniche che possono essere basate su dati storici. Nasce, quindi, l'esigenza di tradurre la lista di eventi predittivi in mansioni richieste e tale processo prende il nome di modellazione della domanda. Tre sono le categorie di eventi su cui si può basare la domanda del personale:

- *Task based demand*: in questo caso la domanda è ottenuta dalle liste delle singole attività che devono essere eseguite. Le attività sono di solito definite

da un tempo di inizio e durata o da una finestra temporale entro cui l'attività deve essere completata, e dalle competenze richieste per eseguire l'attività. Il rostering con la domanda basata sui task viene comunemente impiegato nella logistica di trasporto.

- *Flexible demand*: in questo caso la probabilità di eventi futuri non è ben nota e deve essere modellata usando delle tecniche di previsione. Le richieste di servizio hanno frequenze di arrivi casuali e possibilità di tempi di servizio casuali. Il legame tra le richieste di servizio e il livello di personale richiesto per svolgere la richiesta spesso prende la forma di uno standard di servizio indirizzato ad essere inserito in una coda di analisi della distribuzione degli arrivi. Il risultato è una specifica del numero di persone richieste in tempi diversi del giorno per ogni giorno nell'orizzonte temporale. Una volta generato il modello della *demand flexible*, lo si utilizza per ripartire i turni in modo da poter coprire la domanda. Il modello di domanda flessibile è spesso utilizzato nei call center e nel personale di terra degli aeroporti.
- *Shift based demand*: in questo caso la domanda è ottenuta direttamente dalla specifica del numero di persone che sono richieste per svolgere la richiesta in un determinato turno. La domanda basata sui turni è spesso applicata come una semplificazione di lavoro del rostering basato su domanda o del rostering basato su domanda flessibile. In questa categoria di domande rientrano i servizi di ambulanze dove la richiesta del personale scaturisce dalla necessità di soddisfare la qualità del servizio

Nella seconda fase del processo viene determinato come intervallare i giorni di riposo con i giorni di lavoro nella pianificazione dei turni. Tale problema si presenta

quando la prima fase del rostering è impostata sulla domanda flessibile o basata su turni.

Nella terza fase di *shift scheduling* ci si occupa della scelta, tra un gruppo potenzialmente vasto di candidati, di quali turni devono essere lavorati, insieme ad un'assegnazione del numero di persone per ogni turno, al fine di coprire tutte richieste di attività. Quando il rostering è basato sulla domanda flessibile bisogna considerare l'orario lavorativo e le pause pranzo nei limiti previsti da contratto di lavoro. Nel rostering con domanda basata sui task, il principale compito è quello di trovare un insieme di turni lavorativi che sia in grado di coprire tutti i task.

La quarta fase del processo di rostering *line of work construction* conduce alla determinazione di sequenze di blocchi (*duty* o *stints*, turni tipici) da assegnare ad ogni membro della squadra in un orizzonte temporale. Tale sequenza è fortemente influenzata dal tipo di blocco, ovvero se i blocchi sono turni allora possono essere assegnati giornalmente ai membri della squadra se invece sono *duty* (giornate di lavoro) esse sono il risultato di attività che possono prendere solo una parte di un turno o possono estendersi su più turni e che quindi possono essere inserite in un solo roster. Infine, gli *stints* sono sequenze predefinite di turni di lavoro e giorni di riposo. Essi possono essere il risultato di una aggregazione di attività durante la modellizzazione della domanda o semplicemente dei modelli predefiniti che riflettono norme e regolamenti sul lavoro.

Nella definizione della sequenze di blocchi di turni su un orizzonte temporale bisogna tenere in considerazione una serie di obiettivi e vincoli in conflitto tra loro. Diversi sono i modelli di *line of work*:

- *Cycliacyclic rosters*: in un roster ciclico tutti i membri dello stesso gruppo eseguono esattamente le stesse attività, ma con differenti tempo di inizio per

il primo turno. Questo tipo di roster si applica in situazioni con modelli di domanda ripetitivi. Nei roster aciclici le attività che ogni membro svolge sono completamente indipendenti. Tale tipo di roster si applica in situazioni dove i turni hanno durate e tempo di inizio differente.

- *Stint based*: in alcune aziende solo certe sequenze di turni, chiamate *stint*, possono essere eseguite. La determinazione delle sequenze di lavoro è regolata da norme che indicano come le transizioni di stint possono essere eseguite. Una matrice di transizione di stint può essere usata per specificare quali stint possono essere seguiti un particolare stint. Una tipica applicazione di rostering basato su stint è la pianificazione dei turni degli infermieri.
- *Line of work constraints*: in molte applicazioni i roster flessibili sono significativamente vincolati da regole organizzative e quindi i piani di lavoro vengono eseguiti per un individuo. Tali regole possono imporre il numero di turni di notte sequenziali che possono essere svolti oppure vincoli sul numero di ore lavorate giornalmente.

Nella fase *task assignment* può essere necessario assegnare uno o più task da eseguire nei turni. Questi task possono richiedere particolari capacità o livelli di anzianità e devono perciò essere associati a particolari roster.

L'ultima fase del processo di rostering, chiamata *staff scheduling*, assegna le persone della squadra alle *line of work*.

In letteratura esistono numerosi metodi e tecniche per la risoluzione del problema di turnazione del personale. Essi vengono solitamente suddivisi in base agli obiettivi reali che si vogliono ottenere ovvero in metodi esatti e non esatti. Appartengono alla prima categoria tutti quei metodi che conducono ad una soluzione ottima, utilizzando un approccio di enumerazione implicita, sia esso basato su una formulazione

di programmazione lineare o sfruttando delle proprietà combinatorie del problema. Una strada diversa, che può diventare l'unica percorribile se le dimensioni e/o la complessità del problema sono elevate, è quella che consiste nel cercare soluzioni di tipo euristico, ovvero ottenute applicando un algoritmo concepito in modo da produrre soluzioni che si sperano buone, ma senza garanzia a priori sulla vicinanza all'ottimo.

Negli approcci che utilizzano la programmazione matematica, i problemi di turnazione vengono formulati come modelli di programmazione lineare o programmazione lineare intera o programmazione matematica generale.

Il modello di programmazione più comunemente utilizzato per risolvere i problemi di turnazione è la formulazione Dantzing (vedi [6]) *set covering* o le sue varianti.

Sebbene gli algoritmi basati sulla programmazione ottengono soluzioni a costo più basso, non sono in grado di risolvere in modo efficiente problemi di turnazioni ma possono talvolta essere utilizzati per risolvere versioni semplificate del problema del mondo reale.

Un altro metodo presente in letteratura appartenente alla categoria dei metodi esatti è il *Constraint Programming* o CP. Tale metodo fornisce un potente strumento per trovare soluzioni ammissibili a molti problemi di turnazione in cui è difficile modellare le regole complesse con equazioni matematiche. Si usa tale tecnica quando il problema è fortemente vincolato e/o quando ogni soluzione ammissibile è sufficiente anche se non ottimale. Il vantaggio di questo approccio è che i vincoli complessi possono essere modellati in modo relativamente semplice anche se risulta poco probabile che questa tecnica ottenga buone soluzioni, più o meno ottimali in una vasto numero di soluzioni. Gli approcci CP sono stati applicati alla turnazione del personale ospedaliero (*nurse rostering*), così come descritto in [7].

Data la poca efficienza che deriva dall'uso delle tecniche di CP, spesso vengono

utilizzate in combinazione con altre tecniche tradizionali della ricerca operativa.

Un'importante classe di metodi per risolvere problemi di ottimizzazione complessi è rappresentata dalle tecniche euristiche.

Un algoritmo euristico (o, semplicemente un'euristica) deve essere in grado di produrre una soluzione in tempo relativamente breve. Mentre chiaramente è possibile progettare euristiche specifiche per qualunque problema di ottimizzazione combinatoria, negli ultimi anni hanno acquisito importanza via via maggiore alcuni approcci euristici di tipo generale, detti *metaeuristiche*.

Gli approcci metaeuristici possono essere visti come generalizzazioni di un unico approccio fondamentale, che è quello della *ricerca locale*. La ricerca locale si basa su quello che è, per certi versi, l'approccio più semplice e istintivo dell'ottimizzazione: andare per tentativi e quindi partendo da una soluzione iniziale casuale, si cerca di migliorarla iterativamente ad ogni passo dell'algoritmo.

Euristiche e metaeuristiche vengono spesso utilizzate per risolvere i problemi di turnazione del personale.

Cipriano et al. in [16], per risolvere il problema di rostering in uno scenario ospedaliero, propongono due tecniche ibride che sfruttano le combinazioni delle potenzialità offerte dalla programmazione con vincoli e dalla ricerca locale:

- si effettua un algoritmo di ricerca sistematica basato sulla programmazione con vincoli, migliorato con l'inserimento della ricerca locale in qualche punto del processo di ricerca, ad esempio:
 - per restringere la lista dei figli da esplorare di un nodo dall'albero di ricerca;
 - quando si raggiunge una foglia dell'albero di ricerca per migliorare la soluzione ottenuta, completa o parziale che sia.

- si esegue un algoritmo di ricerca locale, e si usa la programmazione con vincoli come supporto alla ricerca locale, ad esempio:
 - per analizzare il vicinato e scartare quei vicini che non soddisfano i vincoli del problema;
 - per esplorare il vicinato relativo a un frammento della soluzione corrente.

Utilizzare la programmazione con vincoli per trovare una prima soluzione ammissibile è molto vantaggioso rispetto alla generazione di una prima soluzione in modo casuale. L'utilizzo della programmazione con vincoli permette di trovare in tempi brevi una soluzione in cui i vincoli principali del problema risultano soddisfatti, dalla quale il processo di ricerca locale può partire.

L'idea degli autori è di ottenere una prima soluzione ammissibile tramite programmazione con vincoli, per poi migliorare la soluzione ottenuta con tecniche di ricerca locale che esplorino lo spazio delle soluzioni alla ricerca di vicini migliori.

Il sistema proposto è costituito da due moduli principali:

- il modulo *FirstSolution* prende in input un istanza del problema specificata dall'utente e inizia a processare finchè non trova una soluzione fattibile. Se una soluzione fattibile non esiste questo modulo restituisce un errore e si ferma l'esecuzione o, se possibile, si rilassano parti del modello per permettere all'elaborazione di giungere al termine, segnalando comunque all'utente i problemi riscontrati.
- il modulo *LocalSearch* implementa un insieme di algoritmi di ricerca locale per il problema di rostering ospedaliero. Questo modulo prende in input una soluzione fattibile ottenuta dal modulo *FirstSolution* e li migliora per mezzo di una ricerca locale che può essere scelta dall'utente. Le tecniche di ricerca locale

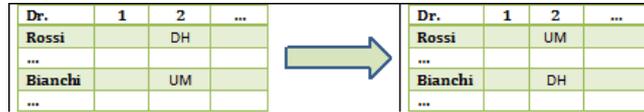


Figura 1.2: Esempio di mossa

implementate in questo modulo sono l'Hill Climbing, lo Steepest Descent e il Tabu Search.

Per realizzare gli algoritmi di ricerca locale è stato definito il concetto di mossa (Fig.1.2) che permette lo spostamento da un elemento dello spazio delle soluzioni ad un altro ad esso vicino come lo scambio di turni di due risorse operative relative allo stesso giorno e alla stessa fascia oraria.

- *Hill Climbing (HC)*: la strategia Hill Climbing sviluppata dagli autori consiste nel generare una mossa di scambio in modo casuale e applicarla alla soluzione corrente.

Se la nuova soluzione migliora o mantiene invariata la funzione di costo senza violare alcun vincolo, allora questa diventa la nuova soluzione corrente; se invece la nuova soluzione peggiora la funzione di costo oppure viola qualche vincolo, si ritorna alla soluzione di partenza. Questo procedimento viene iterato fino allo scadere del tempo stabilito dall'utente.

L'unico aspetto cruciale di questo metodo riguarda la generazione casuale delle mosse, che ha il rischio di generare molte mosse di scambio inutili ovvero mosse che non creano alcun miglioramento al raggiungimento dell'ottimo.

- *Steepest Descent (SD)*: la strategia Steepest Descent consiste nell'enumer-

are tutte le possibili mosse applicabili alla soluzione corrente, calcolare la funzione obiettivo che si otterrebbe applicando ciascuna mossa e poi applicare la mossa che comporta il maggior decremento della funzione obiettivo.

Il processo viene iterato fino a quando alla soluzione corrente non si può applicare alcuna mossa che migliori la funzione obiettivo.

Rispetto all'Hill Climbing risulta più dispendioso individuare la mossa da applicare, però si individuano generalmente mosse che migliorano notevolmente la funzione obiettivo.

L'aspetto chiave di questo metodo è il procedimento di enumerazione delle mosse: poichè calcolare il miglioramento della funzione obiettivo è un'operazione molto dispendiosa computazionalmente, bisogna evitare di eseguirla alla cieca su tutte le possibili mosse.

Molte mosse portano alla violazione di vincoli e quindi non serve calcolare la funzione obiettivo risultante dalla loro applicazione, perchè queste mosse non saranno mai selezionate.

La strategia di enumerazione prodotta contiene quindi una certa conoscenza dei vincoli ed evita di enumerare mosse che li violerebbero, portando così ad un notevole risparmio del tempo di computazione.

- *Tabu Search (TS)*: questo metodo, data una soluzione corrente, esplora una parte delle possibili mosse applicabili, e applica quella che porta al minor valore della funzione di costo, indipendentemente dal fatto che il valore sia minore o maggiore di quello precedente.

Questo permette di uscire dai minimi locali, ma porta al rischio di ciclare continuamente in un insieme di soluzioni. Per evitare viene usata la cosidd-

detta “tabu list”, una lista che contiene le mosse recentemente applicate e proibisce l’applicazione di mosse che risultino inverse di quelle in essa contenute e che quindi riporterebbero la ricerca a visitare uno stato già considerato precedentemente.

- *Hybrid steepest descent* (HSD): questo algoritmo ibrido è guidato dalla strategia steepest descent. L’idea di questo algoritmo è di esplorare i frammenti di vicinato della soluzione corrente ottenuta dalla soluzione della programmazione vincolata. I frammenti di vicinato vengono scelti in modo da formare una partizione di tutto il vicinato della soluzione attuale. L’algoritmo quindi accetta la migliore tra le soluzioni rappresentative che migliorano la funzione obiettivo, ispirato alla strategia steepest descent.

Gli autori hanno effettuato degli esperimenti allo scopo di confrontare i quattro diversi metodi di ricerca locale implementati per la risoluzione del problema (Hill Climbing, Steepest Descent, Tabu Search, Hybrid Steepest Descent).

Il metodo Hill Climbing, nel caso ospedaliero, risulta essere il migliore su tutte le tipologie di istanze: infatti è quello che fa scendere la funzione obiettivo nel modo più veloce.

Questo è dovuto al fatto che il tempo per trovare una mossa migliorativa è molto basso, in quanto non si effettua un’esplorazione esaustiva del vicinato, ma ci si accontenta di spostarsi nel primo vicino generato casualmente che migliora la funzione obiettivo o la lascia immutata; inoltre, grazie a questa caratteristica, l’Hill Climbing non si blocca mai in minimi locali, ma continua a perturbare la soluzione corrente con nuove soluzioni dello stesso valore e quindi prima o poi riesce ad individuare una mossa migliorativa.

Il metodo Steepest Descent classico risulta invece il peggiore: infatti più lento

dell'Hill Climbing, in quanto impiega molto tempo nell'esplorazione del vicinato prima di decidere quale mossa effettuare, e il tempo impiegato nell'esplorare il vicinato non è compensato da un sostanziale decremento della funzione obiettivo ad ogni passo. Inoltre presenta il limite intrinseco di bloccarsi nei minimi locali, e questo comporta che dopo un po' di miglioramenti, al primo minimo locale non riesce più proseguire.

Il Tabu Search è sicuramente migliore dello Steepest Descent: nelle prime fasi delle elaborazioni i metodi si equivalgono (infatti nel grafico si nota che le due linee sono inizialmente sovrapposte), poi entra in gioco il meccanismo di proibizione del Tabu Search che evita di cadere nei minimi locali e consente di continuare a esplorare il vicinato e a migliorare la soluzione.

Il metodo Hybrid Steepest Descent merita un'analisi approfondita, in quanto il suo comportamento varia in base alla dimensione delle istanze su cui viene applicato. Nelle istanze più piccole (10 giorni) il suo andamento è il peggiore in termini di velocità di decremento della funzione obiettivo: questo è dovuto all'alto overhead computazionale di cui questo metodo soffre per esplorare in modo esaustivo tutte le porzioni di vicinato. Sulle istanze piccole questo overhead non viene pagato da un alto decremento della funzione obiettivo per mossa. Nelle istanze medie (20 e 30 giorni) lo sforzo computazionale (in termini di tempo) dell'esplorazione delle porzioni di vicinato viene compensato da decrementi notevoli della funzione obiettivo e questo metodo risulta subito più competitivo del Tabu Search, già dopo le prime decine di secondi.

Questo metodo però scala male sulle istanze grandi (40 giorni): pur mantenendo la sua tendenza a rivelarsi più competitivo del Tabu Search, soffre dell'alta quantità di tempo necessaria all'esplorazione delle porzioni di vicinato, e quindi riesce a superare effettivamente l'andamento del Tabu Search (comunque non in modo netto)

solo dopo qualche centinaio di secondi.

Il problema di rostering e le tecniche di risoluzione proposte precedentemente rientrano in una delle classi di problemi da considerare nella definizione del metodo di risoluzione del problema oggetto di studio. Oltre, all'associazione dei turni e giornate di lavoro al personale (rostering) bisogna tener conto anche dell'associazione ottima di n persone a m interventi, in modo tale che ad ogni intervento sia associato una persona, che in letteratura viene definito come task assignment problem.

Nel seguito del capitolo verranno descritti i modelli già presenti in letteratura studiati ai fini della definizione del modello di schedulazione proposto in questo lavoro.

1.2 La schedulazione del personale in letteratura

Negli ultimi 50 anni molti autori si sono cimentati nello studio del suddetto problema. Negli anni '60 molte pubblicazioni inerenti tale problema erano prettamente orientate alla matematica e principalmente focalizzate sull'affidabilità. Tra il 1970 ed il 1980 invece le pubblicazioni furono più orientate sull'ottimizzazione delle policy di manutenzione. Questo fu un passo avanti, anche se questi modelli erano ancora troppo focalizzati sulla trattabilità matematica piuttosto che su assunzioni ed ipotesi realistiche. Tutto ciò ha causato uno spiacevole divario tra accademici e professionisti. Gli accademici avevano il sentore che l'industria ed i servizi non fossero ancora "pronti" per i loro lavori, mentre i professionisti si sentivano frustrati poiché tali modelli erano ancora troppo teorici. Fortunatamente questa situazione iniziò a cambiare. Attualmente gli accademici pongono più attenzione ai requisiti realistici dei loro lavori ed i professionisti hanno scoperto l'utilità del lavoro accademico.

Nella stesura di questo lavoro di tesi si è fatto riferimento ai lavori di diversi autori

che si sono cimentati finora nella risoluzione di problemi di scheduling degli interventi di manutenzione. In particolare Brucker et al. [9] hanno investigato un modello generale per la risoluzione del *Personnel Scheduling*. Esso può essere formulato nel modo seguente. Sia dato un orizzonte temporale $[0, T]$ diviso in intervalli $[t, t+1[$ per $t = 0, 1, \dots, T-1$. All'interno di questo orizzonte temporale devono essere eseguiti m task. Definiamo $D_j(t)$ il numero di lavoratori necessari per il task j nell'intervallo temporale $[t, t+1[$ ($t = 0, 1, \dots, T-1$). Dato un insieme E di n lavoratori, ad ogni lavoratore $e \in E$ è associato un sottoinsieme Q_e di task per cui e risulta qualificato. Quindi, un pattern di lavoro (turno) per un lavoratore e è definito da:

- un vettore binario $(w_e(t))_{t=0}^{T-1}$ dove $w_e(t) = 1$ se e solo se il lavoratore e è disponibile nel periodo $[t, t+1[$;
- un assegnamento di task appartenente a Q_e per ogni periodo temporale $[t, t+1[$ con $w_e(t) = 1$.

A ciascun impiegato è associato un insieme di operazioni (mansioni) che esso può svolgere in base alle proprie competenze tecniche (skill).

Pertanto, la soluzione del modello sarà un insieme di pattern di lavoro (turni), ciascuno associato ad uno specifico impiegato. Ovviamente, nel caso in cui un impiegato possa svolgere diverse mansioni, un pattern di lavoro specificherà, per ogni periodo di tempo, la mansione che esso andrà a svolgere.

I pattern di lavoro sono rappresentati da vettori binari $\pi = (\pi(j, t))$ dove $(\pi(j, t) = 1)$ se e solo se $[t, t+1[$ è un periodo di lavoro nel quale il task j è stato eseguito.

La Fig.1.3 mostra un esempio di pattern di lavoro per un impiegato che è in grado di svolgere due tipologie di mansioni (task 1 e task 2). Assumiamo di avere un orizzonte temporale di 7 giorni e che $t = 0$ corrisponda a Lunedì. Inoltre, assumiamo che un impiegato e , qualificato per svolgere i due task, lavori al task 1 nei giorni

	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato
task 1	1	0	1	1	0	1
task 2	0	1	0	0	1	0

Figura 1.3: Esempio di pattern di lavoro

Lunedì, Mercoledì, Giovedì e Sabato; mentre, nei restanti, si occuperà della mansione 2.

Quindi il corrispondente pattern di lavoro π sarà definito come $\pi(1, v) = 1$ per $v = 0, 2, 3, 6$, $\pi(2, v) = 1$ per $v = 1, 4$ e $\pi(j, t) = 0$ per tutti gli altri valori (j, t) .

Denotiamo con P_e l'insieme di tutti i possibili pattern fattibili per un impiegato e , dove la fattibilità di un pattern è specificata da una serie di vincoli forti che dipendono dallo specifico problema.

Si deve assegnare agli impiegati $e \in E$ un pattern di lavoro fattibile $\pi_e \in P_e$ tale che:

- la domanda $D_j(t)$ di tutti i task j è coperta in ogni periodo $[t, t + 1[$;
- il costo dell'assegnazione agli impiegati sia minimizzato.

Tra tutti i possibili pattern di lavoro assegnabili ad un impiegato, verranno presi in considerazione solo ed esclusivamente quelli che soddisfano specifici requisiti e che in genere sono il risultato della risoluzione di modelli matematici (per gli approcci esatti) o di procedure euristiche (per gli approcci approssimati).

Uno degli obiettivi che in genere si tende ad ottimizzare nella definizione di modelli matematici è sicuramente il costo complessivo.

Introdotta la variabile $x_{e\pi}$ che assume valore 1 se e solo se il pattern di lavoro π è assegnato all'impiegato e e indicato con $c_{e\pi}$ il costo del pattern di lavoro π assegnato all'impiegato e , il problema può essere modellato nel seguente modo:

$$\min \sum_{e \in E} \sum_{\pi \in P_e} c_{e\pi} x_{e\pi} \tag{1.1}$$

soggetto a

$$\sum_{\pi \in P_e} x_{e\pi} \leq 1, \quad e \in E \tag{1.2}$$

$$\sum_{e \in E} \sum_{\pi \in P_e} \pi(j, t) x_{e\pi} \geq D_j(t), \quad \forall(j, t) \tag{1.3}$$

$$x_{e\pi} \in \{0, 1\} \quad \forall e \in E \quad e \quad \pi \in P_e$$

La funzione obiettivo (1.1), da minimizzare, rappresenta il costo complessivo di assegnazione degli impiegati ai vari pattern di lavoro. Il vincolo (1.2) assicura che ad ogni impiegato sia assegnato al più un pattern di lavoro. Il vincolo (1.3) assicura che il numero di impiegati assegnati al task j nel periodo $[t, t + 1[$ sia almeno uguale alla domanda di j nel periodo $[t, t + 1[$. Diverse possono essere le particolarizzazioni del modello (1.1)-(1.3); ad esempio, si può prevedere la minimizzazione/massimizzazione di una misura di performance differente dal costo totale e/o l'introduzione di altri vincoli operativi.

Una classica applicazione è la determinazione di schedulazioni dei turni di lavoro degli infermieri, in cui: l'orizzonte di pianificazione è, tipicamente, costituito da tutti i giorni della settimana e ci sono soltanto 4 tipi differenti di mansioni da eseguire.

Tuttavia, sebbene il modello matematico (1.1)-(1.3) è una formulazione lineare intera a singolo obiettivo, nei casi reali, la sua dimensione potrebbe diventare intrattabile per i risolutori open source (ad esempio Cplex), a causa del fatto che il numero di schedulazioni ammissibili risulta in genere considerevolmente elevato, così come gli impiegati coinvolti e le mansioni da svolgere. Altri approcci euristici e

meta-euristici per risolvere il problema in questione su grandi dimensioni sono presenti in letteratura, come quello presentato da Dowsland in [10] per i problemi degli infermieri. L'approccio proposto è basato su tabu search ed è ideato per esplorare tutti i turni fattibili predefiniti per gli infermieri con l'obiettivo di trovare la migliore pianificazione che soddisfa i requisiti di copertura.

Scopo, però, di questo progetto di tesi è di estendere il problema a singolo obiettivo ad un framework più generale che preveda l'ottimizzazione simultanea di diverse misure di performance.

Un numero limitato di meta-euristiche è stato applicato ai problemi multi-obiettivo di scheduling del personale. Tuttavia presentiamo una breve descrizione di quelli più significativi tra cui quello studiato da Panta et al. in [11], che tratta dei cosiddetti problemi di *Airlines Crew Rostering*, relativi all'assegnazione degli equipaggi ai voli previsti da una compagnia aerea. Il carico di lavoro di ciascun equipaggio (massimo numero di ore di volo, massimo numero di decolli, etc.) è regolamentato dalle leggi vigenti in uno specifico Paese. Ciascuna compagnia aerea può comunque definire dei propri limiti interni: alcune, ad esempio, riferiscono i vincoli sul carico di lavoro ad un singolo giorno; altre, invece, li riferiscono ad un intero periodo di pianificazione (tipicamente un mese). I piloti della maggior parte delle compagnie aeree nel mondo vengono assegnati per motivi di sicurezza ad un solo tipo di velivolo (la tendenza è quella di adeguare il pilota il più possibile a un certo tipo di aeromobile). Pertanto, quando una flotta ha diversi tipi di aerei, gli orari mensili sono in primo luogo pianificati per i piloti di un tipo di aeromobile, poi per i piloti di un altro tipo di aeromobile, etc. A differenza dei piloti, le hostess e gli steward più spesso hanno competenze per lavorare su diverse imbarcazioni. Quindi, in pratica il personale di cabina può essere assegnato in due possibili modi. Il primo prevede l'assegnazione del personale, così come avviene per i piloti, ad ogni aeromobile. Il secondo modo

assegna simultaneamente tutto personale di bordo a rotazione su tutti i tipi di aeromobili. In questo caso, le dimensioni del problema diventano abbastanza grandi. Il lavoro presentato dagli autori divide i piloti in gruppi in accordo al tipo di aeromobile per i quali hanno una licenza e risolve il problema della turnazione del personale aereo nell'ambito del gruppo di piloti che volano sullo stesso tipo di aeromobile.

Il problema affrontato può essere formulato nel modo seguente: per una determinata schedulazione aerea, assegnare equipaggi, in modo tale che la funzione obiettivo, regolamentata dalla specifica organizzazione aerea, sia massimizzata (o minimizzata) rispettando contemporaneamente tutti i vincoli che regolano le operazioni di equipaggio in modo da non mettere in pericolo la sicurezza del volo. Alcuni vettori aerei hanno come obiettivo la riduzione dei costi di equipaggio totali, altri cercano di massimizzare l'utilizzo dell'equipaggio e minimizzare il numero di membri dell'equipaggio di riserva, etc.

Riportiamo la formulazione matematica per il suddetto problema facendo le seguenti osservazioni:

- il tempo di volo cumulativo per un pilota non può superare le 85 ore;
- il numero di decolli durante il mese non può essere maggiore di 90;
- il numero di ore di lavoro durante il mese non può essere superiore a 160;
- i piloti devono avere un giorno libero dopo 5 giorni consecutivi di lavoro.

Denotiamo con m il numero totale di piloti che devono essere assegnati ai turni di lavoro durante il mese e con k il numero totale di rotazioni che devono essere eseguite.

Definiamo le seguenti variabili binarie:

$$\begin{aligned}
 p_{il} &= \begin{cases} 1, & \text{se il pilota } i \text{ può trascorrere la giornata a lavoro} \\ 0, & \text{altrimenti} \end{cases} \\
 i &= 1, \dots, m \\
 l &= 1, \dots, 30 \\
 q_{il} &= \begin{cases} 1, & \text{se la rotazione } j \text{ parte il giorno } l \text{ del mese} \\ 0, & \text{altrimenti} \end{cases} \\
 j &= 1, \dots, k \\
 l &= 1, \dots, 30
 \end{aligned}$$

Il numero totale di possibili giorni di lavoro P di tutti gli m piloti durante il mese seguente è dato da:

$$P = \sum_{i=1}^m \sum_{l=1}^{30} p_{il} \quad (1.4)$$

Denotiamo con d_j la lunghezza della rotazione j espressa come tempo di volo. La lunghezza totale D di tutte le k rotazioni fatte nel mese seguente è:

$$D = \sum_{j=1}^k d_j. \quad (1.5)$$

Denotiamo con a la media giornaliera di volo di ogni pilota m :

$$a = \frac{D}{P} = \frac{\sum_{j=1}^k d_j}{\sum_{i=1}^m \sum_{l=1}^{30} p_{il}}. \quad (1.6)$$

La situazione ideale, per risolvere il problema di rostering, sarebbe di avere per ognuno degli m piloti un tempo medio di volo giornaliero uguale ad a . Denotiamo con a_i^* il tempo di volo mensile ideale per ogni pilota i . Questo è dato da:

$$a_i^* = a \sum_{l=1}^{30} p_{il}, \quad , i = 1, \dots, m. \quad (1.7)$$

Da notare che tale tempo risulta variabile da pilota a pilota poichè può cambiare il numero totale di giorni nel mese che un piltoa può essere assegnato a un lavoro.

Introduciamo la variabile binaria x_{ij} così definita:

$$x_{ij} = \begin{cases} 1, & \text{se il pilota } i \text{ è assegnato alla rotazione } j, \\ 0, & \text{altrimenti.} \end{cases}$$

Denotiamo con a_i il tempo reale di volo mensile che il pilota i ha volato:

$$a_i = \sum_{j=1}^k d_j x_{ij}, \quad i = 1, \dots, m. \quad (1.8)$$

Sia F_1 la deviazione media tra il tempo di volo mensile reale e quello ideale elevato alla potenza r :

$$F_1 = \frac{1}{m} \sum_{i=1}^m \left| \frac{\sum_{j=1}^k d_j x_{ij} - a \sum_{l=1}^{30} p_{il}}{a \sum_{l=1}^{30} p_{il}} \right|^r. \quad (1.9)$$

I modelli che minimizzano la deviazione come obiettivo sono di solito vulnerabili a soluzioni che includono una deviazione molto grande. Sebbene matematicamente flessibile, nella realtà una tale soluzione può essere considerata non fattibile.

E' chiaro che più piccolo è il valore di F_1 e più il tempo mensile reale di volo si avvicina al quello ideale. Il valore di F_1 può servire come uno delle funzioni obiettivo quando risolviamo il problema di rostering.

Il secondo obiettivo definito dagli autori mette in relazione la deviazione media assoluta (per pilota) tra il numero reale e ideale di possibili indennità straniere giornaliere consentite durante il mese.

Denotiamo con b la parte ideale giornaliera consentita che ogni pilota dovrà

ricevere durante il mese:

$$b = \frac{\sum_{j=1}^k c_j}{m \sum_{l=1}^{30} p_{il}}, \quad (1.10)$$

dove c_j è uguale al numero totale di indennità giornaliere consentite nella rotazione j .

Introduciamo b_i^* il numero mensile ideale di indennità giornaliere consentite al pilota i :

$$b_i^* = \left[b \sum_{l=1}^{30} p_{il} \right], \quad i = 1, \dots, m. \quad (1.11)$$

dove $[y]$ è il numero reale y arrotondato al numero intero più vicino.

Indichiamo con F_2 la deviazione media assoluta (per pilota) tra il numero di indennità giornaliere ideali e reali consentite durante il mese elevata alla potenza di r :

$$F_2 = \frac{1}{m} \sum_{i=1}^m \left| \frac{\sum_{j=1}^k c_j x_{ij} - \left[b \sum_{l=1}^{30} p_{il} \right]}{\left[b \sum_{l=1}^{30} p_{il} \right]} \right|^r. \quad (1.12)$$

Quando risolviamo il problema di rostering, ci si focalizza sulla minimizzazione di F_2 .

Le funzioni obiettivo descritte sono soggette a vincoli che devono essere soddisfatti. Come già detto precedentemente i vincoli considerati dagli autori sono:

- *Vincolo sul tempo di volo*

Le ore totali di volo accumulate durante il mese non possono essere superiore a 85:

$$\sum_{j=1}^k d_j x_{ij} \leq 85, \quad i = 1, \dots, m.$$

- *Vincolo sul numero totale di decolli durante il mese*

Il numero totale di decolli durante il mese non può essere superiore a 90:

$$\sum_{j=1}^k v_j x_{ij} \leq 90, \quad i = 1, \dots, m,$$

dove v_j è il numero di decolli compresi in j rotazioni.

- *Vincolo sul numero totale di ore lavorate in un mese*

Il numero totali di ore lavorate in un mese per pilota non può essere superiore a 160:

$$\sum_{j=1}^k D_j x_{ij} \leq 160, \quad i = 1, \dots, m,$$

dove D_j rappresenta il numero di ore necessarie per completare la j - *esima* rotazione.

- *Vincolo sulle limitazioni del giorno libero*

Ogni pilota deve avere un giorno libero dopo 5 giorni consecutivi di lavoro.

Definita la seguente variabile binaria:

$$q_{jl} = \begin{cases} 1, & \text{se la rotazione } j \text{ parte il giorno } l \text{ del mese,} \\ 0, & \text{altrimenti,} \end{cases}$$

$$i = 1, \dots, m$$

$$l = a, \dots, 30$$

si ha:

$$\sum_{j=1}^k x_{ij} t_j \sum_{l=p}^{p+5} q_{jl} \leq 5, \quad i = 1, \dots, m, \quad p = 1, \dots, 25,$$

dove t_j è il numero totale di giorni richiesti ad un equipaggio aereo per fare la j - *esima* rotazione.

- *Vincoli sulla limitazione del giorno libero nella rotazione*

Mentre si svolge una rotazione, ad un pilota può non essere dato un giorno libero:

$$\sum_{j=1}^k x_{ij} = \sum_{j=1}^k x_{ij} \sum_{l=1}^{30} q_{jl} \prod_{s=l}^{l+t_j-1} p_{is}, \quad i = 1, \dots, m$$

- *Vincolo sulla limitazione di un solo pilota*

Una rotazione può avere un solo pilota:

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, k$$

- *Vincoli sulle sovrapposizioni*

Le rotazioni assegnate ad un pilota non si devono sovrapporre nel tempo.

Denotiamo con T_j l'istante in cui un pilota inizia la rotazione j -esima e con T'_j l'istante in cui è pronto per iniziare una nuova rotazione dopo aver completato la rotazione j -esima. L'intervallo entro il quale un pilota è occupato è definito da $[T_j, T'_j]$. Due rotazioni j e p si sovrappongono se si trovano nello stesso l'intervallo temporale.

Definiamo la seguente variabile binaria:

$$\rho_{rs} = \begin{cases} 1, & \text{se la rotazione } j \text{ si sovrappone a } s \\ 0, & \text{altrimenti} \end{cases}$$

Il vincolo è espresso da:

$$x_{ij} \sum_{s=1}^k \rho_{js} x_{is} (s - j) = 0, \quad i = 1, \dots, m, \quad j = 1, \dots, k.$$

Panta et al indicano come *soluzione soddisfacente* quella soluzione che garantisce a tutti i membri dell'equipaggio di avere un carico di lavoro approssimativamente

uguale. L'algoritmo, per risolvere il problema in esame, rappresenta una modifica dell'euristica *pilota per pilota* ("pilot by pilot", proposta in [12] e in [13]). Essa determina l'assegnazione su base mensile per il primo pilota, poi per il secondo e così via. I piloti sono ordinati in base alla propria anzianità. Tuttavia, potrebbe capitare che questa procedura euristica non sia in grado di distribuire in maniera equa il carico di lavoro. Pertanto, essa è spesso combinata in un framework euristico più complesso, al fine di distribuire equamente il carico fra i dipendenti. L'algoritmo risolutivo è costituito dai seguenti passi:

1. Usare la procedura *pilota per pilota* per generare una soluzione iniziale ammissibile;
2. Usare il *Simulated Annealing* per il problema di ottimizzazione multi-obiettivo al fine di migliorare la soluzione ottenuta al punto precedente.

La procedura euristica *pilota per pilota* è particolarizzata nel modo seguente:

1. Selezionare un membro dell'equipaggio p ;
2. Assegnare p a più turni possibili nel rispetto dei vincoli imposti;
3. Rimuovere p dalla lista e tutti i turni che ad esso sono stati assegnati;
4. Verificare se tutti i membri della lista sono stati assegnati a turni. Se sì, l'algoritmo si arresta; altrimenti, ritorna al passo 1.

Il passo 1 dell'algoritmo seleziona un membro dell'equipaggio a cui verranno assegnati un certo numero di turni. La lista dei membri è preventivamente ordinata in base a qualche criterio, ad esempio, l'anzianità. Qualora questo criterio di ordinamento non dovesse essere performante per lo specifico problema in esame, la scelta del membro può anche avvenire in maniera casuale.

Per quanto, invece, riguarda la procedura di *Simulated Annealing*, prendendo spunto dal framework originariamente proposto da Kirkpatrick ¹, si costruiscono i seguenti step:

1. selezionare casualmente un pilota candidato ad un turno. Usando uno schema di “sottrazione-addizione”, si cerca di assegnare il pilota al turno. Se il turno non può essere assegnato al pilota, a causa di alcuni vincoli operativi, si sceglie casualmente un secondo pilota candidato e così via. Se dopo z tentativi (parametro, tipicamente, indicato dal decisore) nessun pilota è stato determinato, il turno risulterà non assegnato e l’algoritmo proseguirà con lo step 5.

Quando si usa lo schema “sottrazione-addizione” ed il pilota può essere sostituito nel turno in questione, verificare se esso è stato preventivamente eliminato da un turno a cui risultava già assegnato e che la deviazione relativa del tempo di volo mensile da quella ideale (r_1), insieme alla deviazione relativa del numero mensile di indennità diaria estera da quella ideale (r_2), non sia negativa. Se uno dei due valori dovesse essere negativo l’algoritmo prosegue con lo step 2.

2. Quando $r_1 < 0$ oppure $r_1 < 0$ e $r_2 \geq 0$, ordinare tutti i turni precedentemente assegnati al pilota candidato (eccetto il turno in questione), in base alla lunghezza decrescente espressa in ore di volo. Quando $r_1 \geq 0$ e $r_2 < 0$,

¹Kirkpatrick et.al. in [14] hanno ideato una tecnica di *Simulated Annealing* procedendo per analogia sul metodo proposto da Metropolis et al. in [15], i quali hanno proposto un metodo per calcolare la distribuzione di un sistema di particelle all’equilibrio termico. In questo metodo, supposto che il sistema si trovi in una configurazione q avente energia $E(q)$ si genera un nuovo stato r , avente energia $E(r)$, spostando una delle particelle dalla sua posizione: la nuova configurazione viene quindi confrontata con la vecchia. Se $E(r) \leq E(q)$ il nuovo stato viene accettato; se $E(r) > E(q)$ esso non viene respinto, bensì viene accettato con una certa probabilità.



Figura 1.4: Procedura pilota-pilota assegnazione iniziale

ordinare tutti i turni precedentemente assegnati al pilota candidato (eccetto il turno in questione) in base alla lunghezza decrescente espressa in numero di indennità diurne estere. L'algoritmo prosegue con lo step 3.

3. Cercare di assegnare il primo turno dalla lista, formata al passo 2, al pilota che è stato rimosso da un altro turno. Se questa operazione dovesse generare una mossa non ammissibile, a causa dei vincoli operativi, cercare di assegnare il secondo turno della lista, quindi il terzo e così via. Se non dovesse essere possibile assegnare il pilota, l'algoritmo prosegue con lo step 5 (in questo caso la mossa è anche completata usando lo schema di "sottrazione-addizione"). Qualora, invece, si dovesse stabilire che il pilota può essere assegnato ad un turno della lista, l'algoritmo prosegue con lo step 4.
4. Assegnare il pilota che ha avuto un turno nello step **3**; proseguire con lo step 5.
5. Terminare l'algoritmo.

La comprensione dettagliata dei passi può avvenire mediante il seguente esempio esplicativo. Si supponga di aver stabilito gli assegnamenti mostrati in Fig. 1.4, soluzione della procedura euristica "pilota per pilota".

Ciò significa che il primo turno è stato assegnato al pilota $n. 7$; il secondo al pilota $n. 8$ e così via. Quindi, si seleziona, casualmente, un pilota che dovrà essere assegnato al primo turno, al posto del pilota $n. 7$. Si supponga che il pilota scelto

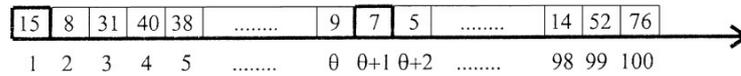


Figura 1.5: Procedura “pilota per pilota” - assegnazione dopo la prima mossa

sia il $n. 15$ e che, dopo attente verifiche, il pilota $n. 15$ possa essere assegnato al primo turno, al posto del pilota $n. 7$. Quindi, si sostituisce il pilota $n. 7$ con il pilota $n. 15$. In tal modo, il pilota $n. 7$ ha un suo tempo complessivo di volo, così come l’indennità diaria estera, ridotto (a causa del fatto che non ha più il turno 1 assegnato). Allo stesso tempo, il pilota $n. 15$ avrà il tempo di volo e il numero di indennità diaria incrementato (a causa della nuova assegnazione). Tipicamente, per evitare questa non uniformità nell’assegnazione, si preferisce effettuare uno scambio equo: il pilota $n.15$ assegnato al turno 1 ed il pilota $n. 7$ al turno $\theta + 1$. In tal caso si ottiene la nuova soluzione mostrata in Fig. 1.5 . Quando una mossa viene applicata, la procedura, comunque, verifica l’ammissibilità turno per turno. Tali mosse di scambio vengono applicate in maniera iterativa fino ad un numero fissato.

Una nuova soluzione, ottenuta dopo aver applicato tante mosse di perturbazione, come sopra specificato, può essere o non essere accettata, poichè possono esserci diverse criteri per accettare la nuova soluzione. Denotiamo con:

- x la soluzione corrente ed y la nuova soluzione ottenuta;
- h il numero di funzioni obiettivo nel problema considerato;
- w_i il peso assegnato a ciascuna funzione obiettivo i ;
- T il valore della temperatura corrente;
- $p_{xy}(T)$ la probabilità di accettare la nuova soluzione y .

Nel problema considerato dagli autori h è uguale a 2 e viene usata una nuova semplice funzione obiettivo:

$$F(f) = \sum_{i=1}^h w_i f_i$$

Pertanto la probabilità di accettare la nuova soluzione proposta è data da:

$$pxy(T) = \min\left\{1, e^{-\frac{w_i(f_i(x) - f_i(y))}{Y}}\right\}.$$

Quando la probabilità è uguale ad 1, la nuova soluzione è considerata migliore di quella corrente. Se la probabilità è minore di 1, si sceglierà un numero casuale u e si effettuerà la medesima procedura del classico approccio del *Simulated Annealing* mono-obiettivo

Lo schema algoritmico risulta quindi essere il seguente:

1. Definire un vettore $ST = (t_1, t_2, \dots, t_q)$ le cui q componenti sono valori diversi di temperature. Tale vettore è ordinato per temperatura decrescente.
2. Utilizzando come soluzione iniziale quella che si ottiene a valle della procedura “pilota per pilota”, poniamo $i = 1$ e $T = t_1$ ovvero uguale alla prima componente di ST .
3. Perturbare casualmente la soluzione.
4. Calcolare la differenza di funzione obiettivo Δ tra la nuova soluzione ottenuta al passo 3 e quella corrente. Se $\Delta > 0$ si va al passo 6; altrimenti al passo 5.
5. Generare un numero casuale u , uniformemente distribuito tra 0 e 1. Calcolare la probabilità di accettare la nuova soluzione P , in base alla definizione data precedentemente. Se $u < P$, si va al passo 6. Se $u > P$, si mantiene la vecchia soluzione e si va al passo 3.

6. Accettare la nuova soluzione e memorizzare il nuovo valore di funzione obiettivo.
7. Se il sistema raggiunge l'equilibrio termico in corrispondenza della soluzione corrente T , settare $i = i + 1$. Se $i > q$, arrestare l'algoritmo. Altrimenti, settare $T = t_i$ uguale all' i -esimo valore di temperatura nel vettore ST e andare al passo 3. Se l'equilibrio termico non è stato raggiunto, senza modificare il valore di T , ritornare al passo 3.

Questo algoritmo ha due loop innestati: in quello più esterno si modifica la temperatura, mentre in quello più interno si determina il numero di perturbazioni da effettuare. Si noti che tale procedura euristica rappresenta un valido framework iniziale da estendere ed applicare al caso di schedulazione preso in esame in questa attività di ricerca. Nel caso specifico, i piloti verrebbero, più propriamente, sostituiti dai tecnici, addetti alla manutenzione delle reti elettriche. I voli presi in esame sarebbero le pianificazioni di lavoro possibili e i vincoli operativi considerati dipenderebbero dallo specifico scenario da risolvere, in base, anche, allo specifico contratto di lavoro. Risulta, quindi, chiaro che tale framework euristico è più generale possibile e che una o più estensioni potrebbero fornire buone soluzioni.

Capitolo 2

Schedulazione degli interventi di manutenzione delle reti elettriche

Nella gestione della schedulazione degli interventi di manutenzione elettrica sorge la necessità di integrare in un'unica soluzione aspetti relativi a problematiche di routing e problemi di scheduling, e il dover rispondere in maniera efficace all'emergenza.

Il problema della schedulazione degli interventi consiste essenzialmente nel dover associare un insieme di risorse operative ad un insieme di task di manutenzione.

Ad ogni task di manutenzione sono associati:

- una priorità che misura, all'interno dell'insieme dei task di manutenzione, l'importanza della tempestività del task;
- una durata che misura il tempo necessario per l'esecuzione del task;

- un insieme di competenze che la squadra deve possedere per poter svolgere un determinato task;
- una posizione geografica dove il task deve essere eseguito.

Il modello proposto deve tener conto, ai fini della schedulazione, di alcuni fattori che influenzano le risorse operative tra cui:

- tempo necessario alla risorsa operativa di raggiungere il luogo dell'intervento, ovvero in base alla posizione di partenza: quanto tempo impiega per raggiungere il luogo in cui sarà svolto il successivo task;
- informazioni relative alla disponibilità, ovvero se impegnata in altri lavori e quindi non disponibile;
- locazione fisica, rappresentata dalla sede dalla quale la risorsa operativa comincia il suo turno di lavoro; tale sede può variare nel tempo, poichè una risorsa operativa può iniziare le sue attività partendo dalla propria abitazione oppure dal luogo in cui è presente l'impianto;
- insieme di competenze che descrivono la competenza della squadra.

Valgono le seguenti assunzioni:

- una squadra è composta da: risorsa operativa ed automezzo;
- la durata dell'intervento è costante, nel senso che non dipende dalla risorsa operativa che è assegnata ad esso;
- una risorsa operativa può lavorare su un solo task per volta;

- una risorsa operativa è indisponibile per altre attività durante l'esecuzione del task cui è assegnata;
- il modello tiene conto che una risorsa operativa deve cominciare e finire la giornata lavorativa presso la sua sede di assegnazione e deve avere sufficiente tempo per spostarsi da e verso tale sede all'inizio ed alla fine della giornata lavorativa;
- il tempo standard impiegato da una risorsa operativa per spostarsi dalla sua posizione di partenza al luogo di esecuzione di un task dipende dalla distanza tra il punto di partenza ed il punto di arrivo; tale valore deve essere corretto prendendo in considerazione parametri che variano dinamicamente nel tempo quali, ad esempio, l'orario in cui ha inizio lo spostamento, parametri che influenzano la velocità dello spostamento (ad esempio condizioni metereologiche, traffico, etc.);
- per interventi urgenti potrebbe accadere che una squadra non sia fornita del materiale necessario: in questo caso al tempo di viaggio calcolato sulla base dell'attuale posizione e di quella dell'intervento va sommato il tempo necessario al prelievo dei materiali.

La schedulazione degli interventi può essere suddivisa in:

- schedulazione ordinaria, quando vengono assegnati tutti i task da eseguire all'interno di un orizzonte temporale predefinito quale ad esempio una giornata lavorativa;
- schedulazione dinamica, quando il sistema deve gestire emergenze non pianificate, spostando eventualmente i task previsti al giorno successivo, deve ricevere le

informazioni relative alla posizione sul campo degli operatori e infine eliminare dalla lista degli interventi quelli già effettuati.

2.1 Informazioni di ingresso

Il modello è caratterizzato dai seguenti insiemi:

- **Q** l'insieme delle squadre a disposizione: il numero totale di squadre a disposizione è Q ($|Q| = Q$). La squadra è costituita da risorse operative e dall'automezzo.
- **C** l'insieme delle possibili competenze dei tecnici: il numero totale di possibili competenze è C ($|C| = C$).
- **I** l'insieme dei possibili interventi: il numero totale di tipologie di intervento è I ($|I| = I$).
- **P** l'insieme delle possibili posizioni geografiche: il numero totale di posizioni geografiche è P ($|P| = P$).
- **O** l'insieme delle posizioni geografiche che identificano le postazioni di partenza (cioè da dove iniziano e terminano il turno di lavoro) delle squadre; il numero totale di tali postazioni è $|O|$ e O_q rappresenta la postazione di partenza della squadra q .
- **H** rappresenta l'orizzonte temporale per il quale si sta risolvendo il problema di scheduling.

Il modello è definito su un grafo completo $G(P, E)$, dove E rappresenta l'insieme degli archi che collegano le varie postazioni di intervento.

Ogni posizione geografica è identificata dalle informazioni sulle coordinate del luogo:

$$p_n \text{ con } 1 \leq n \leq N.$$

Ogni tipologia di intervento è caratterizzata da una distribuzione del tempo di intervento (tempo necessario ad effettuare la manutenzione/riparazione):

$$F_i(t) \text{ con } 1 \leq i \leq I.$$

Il numero totale di interventi da schedulare in un determinato intervallo di tempo (ad esempio una giornata lavorativa) è N .

Inoltre è stato associato ad ogni intervento una priorità che è caratterizzata da due valori: normale o alta.

La lista degli interventi da schedulare è rappresentata tramite una lista di triple che presentano la seguente struttura:

$$\langle \textit{posizione}, \textit{tipologia di intervento}, \textit{livello di priorità} \rangle$$

La lista L degli interventi da effettuare, quindi, è definita nel seguente modo:

$$L = \{(p_1, F_1(t), l_1), \dots, (p_n, F_n(t), l_n), \dots, (p_N, F_N(t), l_N)\} \text{ con } p_N \in P, F_n(t) \in I, l_n \in \{\textit{alta}, \textit{normale}\}.$$

2.2 Formulazione matematica del problema di scheduling

La schedulazione si occupa di stabilire l'ordine e il numero degli interventi che la squadra dovrà affrontare durante l'intera giornata di lavoro.

L'algoritmo di schedulazione considera come elementi predominanti:

- le competenze delle squadra;
- le priorità dei task;
- la posizione geografica dell'intervento;
- i tempi di spostamento.

Il numero di interventi che ogni squadra deve compiere è N_q con $1 \leq q \leq Q$.

Il processo di schedulazione genera una lista ordinata L_q degli interventi che ogni squadra deve compiere.

$$\sum_{q=1}^Q N_q = N$$
$$\bigcup_{q=1}^Q L_q = L$$

Se ci sono interventi che richiedono particolari *competenze* vengono inseriti nelle liste delle squadre che presentano quelle competenze. Di seguito sono elencate le funzioni che associano ad ogni squadra e ad ogni intervento, un insieme di competenze.

$$F_{cq} : Q \rightarrow 2^C,$$
$$F_{ci} : I \rightarrow 2^C.$$

Si ricorda che l'insieme \mathbf{C} rappresenta l'insieme delle possibili competenze.

La funzione F_{cq} associa ad ogni squadra un insieme di competenze ovvero un sottoinsieme di C . La funzione F_{ci} associa ad ogni intervento un insieme di competenze richieste ovvero un sottoinsieme di C .

L'espressione analitica 2^C rappresenta l'insieme delle parti di C ovvero tutti i possibili sottoinsiemi:

- l'insieme vuoto;
- C stesso;
- tutti i sottoinsiemi con un solo elemento;
- tutti i sottoinsiemi di due elementi e così via.

Quindi un intervento i può essere assegnato ad una squadra q se e solo se $F_{ci}(i) \subseteq F_{cq}(q)$.

Ciò significa che la squadra q può svolgere l'intervento i solo se $F_{cq}(q)$ contiene $F_{ci}(i)$, ossia la squadra deve avere tutte le competenze necessarie ad eseguire l'intervento.

Se nella lista L sono presenti interventi a priorità alta questi devono essere inseriti nelle liste L_q nelle prime posizioni.

Tra gli interventi ad alta priorità hanno la maggiore precedenza quelli la cui segnalazione è stata effettuata prima.

Se nel corso della giornata sopraggiungono interventi non previsti ad alta priorità, il processo di schedulazione procede nel seguente modo:

- si esamina lo stato delle squadre;
- se sono disponibili più squadre, si sceglie la squadra che geograficamente è più vicina alla sede del task;
- si eliminano dalla lista L gli interventi già effettuati;
- si inseriscono nella lista L gli interventi sopraggiunti ad alta priorità;
- si rilancia l'algoritmo di schedulazione.

L'algoritmo di schedulazione considera la lista delle posizioni e della durata degli interventi della squadra q (con $1 \leq q \leq Q$):

$$M_q = \{(p_q^1 t_q^1), (p_q^2 t_q^2), \dots, (p_q^i t_q^i), \dots, (p_q^{N_q} t_q^{N_q})\}$$

dove:

- $p_q^i \in P$ è la posizione dell'intervento i –esimo per la squadra q ;
- t_q^i è la durata dell'intervento i –esimo per la squadra q campionata dalla corrispondente distribuzione (dipendente dal tipo di intervento) con $1 \leq i \leq N_q$.

Se p_q^0 è la posizione di partenza della squadra q e $p_q^{N_q+1}$ è la posizione finale alla quale la squadra deve tornare:

- V_q^i è il tempo di viaggio associato all'intervento i –esimo per la squadra q ;
- $V_q^{N_q+1}$ è il tempo di viaggio per il ritorno alla posizione finale;
- $V_q^i = F(p_q^{i-1}, p_q^i)$: V_q^i può essere considerato una variabile aleatoria se si associa una certa varianza (ad es. dovuta al traffico).

Per interventi urgenti potrebbe accadere che una squadra non sia fornita del materiale necessario: in questo caso al tempo di viaggio calcolato sulla base dell'attuale posizione e di quella dell'intervento va sommato il tempo necessario al prelievo dei materiali.

Per ogni squadra che ha una lista di interventi L_q è associata la durata della giornata lavorativa:

- D_q è la durata della giornata lavorativa per la squadra q ;

- $D_q = \sum_{i=1}^{N_q+1} (V_q^i + t_q^i)$ considerando $t_q^{N_q+1} = 0$.

Indicando con D la durata massima della giornata lavorativa, si vuole che $D_q < D$ per ogni squadra. Se questo non è possibile:

- non sarà possibile effettuare tutti gli interventi N durante la giornata;
- si definisce \bar{N} il numero di interventi effettivamente effettuati durante la giornata;
- si inserisce un costo di straordinario.

Per quanto riguarda il costo del viaggio per lo spostamento della squadra è possibile associare un costo (ad esempio si consideri la tabella ACI):

- CV_q^i è il costo di viaggio associato all'intervento i -esimo per la squadra q ;
- $CV_q^i = F(p_q^{i-1}, p_q^i)$.

Si può ipotizzare che ci sia un costo aggiuntivo per ogni squadra la cui durata della giornata lavorativa supera D :

- Se $D_q > D$ si genera un costo di straordinario;
- CS_q è il costo di straordinario per la squadra q ;
- $CS_q = F(D_q, D)$.

L'algoritmo di schedulazione si pone, come obiettivo principale:

- massimizzare la produttività (task schedulati all'interno della finestra temporale predefinita);

- minimizzare i costi operativi (costi del personale e costi degli spostamenti)
- massimizzare la qualità del servizio, in termini di assegnazione ai task di manutenzione della risorsa operativa adeguata.

L'algoritmo di schedulazione cerca di trovare, quindi, le liste L_q tali che:

- $\sum_{q=1}^Q \sum_{i=1}^{N_q} CV_q^i$ sia minimizzato;
- $\sum_{q=1}^Q CS_q$ sia minimizzato.
- \bar{N} sia massimizzato.

Il modello matematico è illustrato di seguito.

$$\min \alpha \sum_{p \in P'} \sum_{r \in P'} \sum_{i \in I} \sum_{t=0}^H x_{prq} y_{pi}^t (\tau_{\min}^i + g_{pr}) + \beta \theta \sum_{p \in P} \sum_{r \in P} \sum_{q \in Q} d_{pr} x_{prq} + \gamma \sum_{q \in Q} t_q^+ \quad (2.1)$$

$$\sum_{q \in Q} z_{pq} = 1, \quad p \in P' \quad (2.2)$$

$$\sum_{p \in P'} z_{pq} \leq |P'| a_q \quad q \in Q \quad (2.3)$$

$$z_{pq} \leq c_{pq}, \quad p \in P', q \in Q \quad (2.4)$$

$$x_{prq} + x_{r pq} \leq 1, \quad p \in P, r \in P, q \in Q \quad (2.5)$$

$$\sum_{q \in Q} z_{pq} \geq y_{pi}^t, \quad p \in P', t = 1 \dots H, i \in I \quad (2.6)$$

$$\sum_{p \in P'} y_{pi}^t \geq \delta_i, \quad t = 1 \dots H, i \in I \quad (2.7)$$

$$x_{prq} + x_{r pq} \leq z_{pq}, \quad p \in P', r \in P', q \in Q \quad (2.8)$$

$$x_{prq} + x_{r pq} \leq z_{rq}, \quad p \in P', r \in P', q \in Q \quad (2.9)$$

$$x_{prq} + x_{rpq} \leq z_{pq} + z_{rq} - 1 \quad p \in P', r \in P', q \in Q \quad (2.10)$$

$$\sum_{h \in P: (p,h) \in E} x_{phq} - \sum_{h \in P: (h,p) \in E} x_{h pq} = 0 \quad p \in P, q \in Q \quad (2.11)$$

$$x_{prq}(s_{qp} + F_{ip} + g_{pr} - s_{rq}) \leq 0, \quad p \in P, r \in P, q \in Q, i \in I \quad (2.12)$$

$$\sum_{p \in P'} x_{o_q p q} = a_q, \quad o_q \in O, q \in Q \quad (2.13)$$

$$\sum_{p \in P'} x_{p o_q q} = a_q, \quad o_q \in O, q \in Q \quad (2.14)$$

$$\sum_{p \in P} \sum_{r \in P} x_{prq}(F_{ip} + g_{pr}) - D_q = t_q^+ - t_q^- \quad q \in Q, i \in I \quad (2.15)$$

$$t_q^+ \leq D_q u_q, \quad q \in Q \quad (2.16)$$

$$t_q^- \leq D_q v_q, \quad q \in Q \quad (2.17)$$

$$u_q + v_q \leq 1, \quad q \in Q \quad (2.18)$$

$$z_{pq} \in \{0, 1\}, \quad p \in P', q \in Q \quad (2.19)$$

$$x_{prq} \in \{0, 1\}, \quad p \in P, r \in P, q \in Q \quad (2.20)$$

$$x_{r pq} \in \{0, 1\}, \quad p \in P, r \in P, q \in Q \quad (2.21)$$

$$a_q \in \{0, 1\}, \quad q \in Q \quad (2.22)$$

$$c_{pq} \in \{0, 1\}, \quad p \in P, q \in Q \quad (2.23)$$

$$u_q \in \{0, 1\}, \quad q \in Q \quad (2.24)$$

$$v_q \in \{0, 1\}, \quad q \in Q \quad (2.25)$$

$$s_{pq} \geq 0, \quad p \in P, q \in Q \quad (2.26)$$

$$t_q^+, t_q^- \geq 0, \quad q \in Q \quad (2.27)$$

$$g_{pr} \geq 0, \quad p \in P, r \in Q \quad (2.28)$$

$$y_{pi}^t \in \{0, 1\} \qquad p \in P', t = 1 \dots H, i \in I \quad (2.29)$$

La funzione obiettivo è influenzata da tre parametri:

- α definisce l'importanza da dare alla competenza legata alla qualità del servizio. Nel problema oggetto di studio si vuole minimizzare il tempo di interruzione del servizio legato ai vari interventi di manutenzione;
- β definisce il peso da assegnare al costo legato agli spostamenti fra le varie sedi di intervento;
- γ permette di impostare il peso legato al costo degli straordinari per ogni singola squadra.

I valori assegnati ai parametri $\alpha, \beta, e \gamma$ possono variare nell'intervallo $[0, 1]$, purchè scelti in modo da non violare il vincolo $\alpha + \beta + \gamma = 1$.

Il parametro θ è una costante tramite la quale si uniforma il contributo legato al costo degli spostamenti simile, per ordine di grandezza, agli altri due; il suo valore è stato impostato a 0.016.

In dettaglio, il significato delle variabili del modello sopra esposto sono:

- z_{pq} è una variabile binaria che vale 1 se la squadra q è assegnata all'intervento nella postazione p ;
- x_{prq} è una variabile binaria che vale 1 se la squadra q effettua il task nella postazione r immediatamente dopo aver eseguito il task nella postazione p ;
- s_{pq} è una variabile che rappresenta l'istante di tempo in cui la squadra q raggiunge la postazione p ;

- t_q^+ è la durata complessiva del servizio effettuato dalla squadra q eccedente D_q ;
- t_q^- è la durata complessiva del servizio effettuato dalla squadra q non eccedente D_q ;
- u_q vale 1 se la durata complessiva del servizio effettuato dalla squadra q eccede D_q , altrimenti vale 0;
- v_q vale 1 se la durata complessiva del servizio effettuato dalla squadra q non eccede D_q , altrimenti vale 0;
- a_q vale 1 se il team q viene attivato, altrimenti vale 0;
- y_{pi}^t vale 1 se, al tempo t , si verifica un guasto che richiede un tipo di intervento i nella postazione p .

Altri dati introdotti nel modello sono:

- d_{pr} rappresenta il costo sostenuto per lo spostamento dalla posizione geografica p alla posizione geografica r ;
- g_{pr} rappresenta il tempo necessario per lo spostamento dalla posizione geografica p alla posizione geografica r ;
- c_{pq} rappresenta il livello dell'abilitazione tecnica posseduto dalla squadra q associato all'intervento da eseguire nella postazione p ;
- τ_{\min}^i indica un tempo minimo di intervento richiesto nel caso si verifichi un guasto di tipologia i .

Nel seguito definiamo in dettaglio vincoli evidenziati nel modello matematico del problema di scheduling, ovvero:

- il vincolo (2.2) obbliga il modello ad assegnare un dato intervento a una sola squadra;
- il vincolo (2.3) indica che un team è operativo se e solo se almeno un task è stato assegnato ad esso;
- il vincolo (2.4) implica che la variabile z_{pq} sia nulla nel caso in cui la squadra q non abbia le competenze per eseguire il task di intervento nella postazione p ;
- il vincolo (2.5) garantisce che un team q possa servire due task consecutivamente in un solo modo, o prima nella postazione p e poi nella postazione r o viceversa;
- il vincolo (2.6) indica l'obbligo di assegnare una squadra di intervento ad un task di tipo guasto;
- il vincolo (2.7) impone che nel modello venga previsto un numero minimo di guasti di tipologia i , in modo da garantire una maggiore robustezza della soluzione;
- i vincoli (2.8 - 2.9 - 2.10) congiuntamente con il vincolo di interezza delle variabili decisionali coinvolte garantiscono che un team si sposti da p a r se e solo se entrambi sono postazioni di intervento assegnate al team stesso;
- il vincolo (2.11) assicura che ogni squadra assegnata ad un task arrivi nel posto in cui il task deve essere svolto e riparta da tale posto dopo aver svolto il servizio;
- il vincolo (2.12) assicura che se una squadra q è assegnata al task nella postazione p e immediatamente dopo al task nella postazione r , il tempo di arrivo in r sia

compatibile con il tempo di arrivo in p , il tempo di svolgimento del servizio nella postazione p e il tempo di spostamento verso r ; essi possono essere linearizzati, in tal caso saranno nella forma: $s_p + F_{ip} + g_{pr} - s_r \leq s_r + M(1 - x_{prq})$;

- i vincoli (2.13 - 2.14) assicurano che, qualora una squadra sia operativa, essa debba partire dalla propria sede e rientrare necessariamente in essa;
- i vincoli (2.15 - 2.16 - 2.17 - 2.18) congiuntamente alla condizione di interezza di u_q e v_q , misurano la durata del servizio complessivo effettuato dalla squadra q rispetto a D_q .

Capitolo 3

Tecniche di schedulazione degli interventi di manutenzione

Le principali tecniche utilizzate nell'ambito della schedulazione ottimale degli interventi di manutenzione sono basate su algoritmi di ottimizzazione multi-obiettivo. Pertanto, nella maggior parte dei casi, il problema della schedulazione richiede il calcolo di un compromesso ottimale tra diversi obiettivi, i principali dei quali sono rappresentati dai costi totali di manutenzione e dall'affidabilità del sistema.

La schedulazione ottimale degli interventi di manutenzione prevede spesso la soluzione di un problema NP-hard, la cui complessità computazionale cresce in modo esponenziale rispetto alla taglia dell'input. Nell'ambito dei sistemi attuali per la produzione di energia elettrica e la manutenzione di reti distribuzione, il modello di programmazione matematica proposto nel capitolo precedente risulta computazionalmente non fattibile.

Per tale motivo, l'attenzione si è focalizzata sull'applicazione di tecniche di ottimizzazione meta-euristiche multi-obiettivo. Tali tecniche non garantiscono il cal-

colo della soluzione ottima ma sono in grado di determinare in tempi ragionevoli una soluzione spesso vicina all'ottimo globale. Più in dettaglio sono state utilizzate le tecniche basate su Evolutionary Computation (EC).

3.1 Tecniche basate su Evolutionary Computation (EC)

Le tecniche basate su Evolutionary Computation sono modellate sui principi di evoluzione biologica e prevedono la sopravvivenza degli individui (soluzioni) migliori.

I tre principali algoritmi appartenenti a questa categoria sono: *genetic algorithms* [1] - [2] - [3], *evolutionary programming* [4] - [5], and *evolution strategies*.

Tutti gli algoritmi si basano sul concetto di una popolazione di individui (che rappresentano possibili soluzioni a un determinato problema), che si sottopongono a operatori probabilistici, quali *mutazioni*, *selezioni* e talvolta a *ricombinazioni* al fine di migliorare il valore di fitness degli individui. Il *fitness* di un individuo riflette il valore della sua funzione obiettivo rispetto ad una particolare funzione obiettivo che deve essere ottimizzata. L'operatore di mutazione introduce innovazione nella popolazione generando variazioni di individui e l'operatore ricombinazione tipicamente esegue uno scambio di informazioni tra individui diversi di una popolazione. L'operatore selezione impone una forzatura sul processo di evoluzione per preferire gli individui migliori che devono sopravvivere e riprodursi quando i membri della generazione $t + 1$ sono selezionati.

Il loop principale di tutti i principali algoritmi appartenenti a questa categoria di tecniche consiste nella ricombinazione, mutazione, valutazione della funzione di fitness e selezione. Tale loop viene iterato un certo numero di volte finchè non si arrivi in un criterio di arresto.

Lo pseudocodice principale contenuto in tutti gli *algoritmi evolutivi* può essere così sintetizzato:

Algoritmo

```
 $t := 0;$   
initialize  $P(t)$ ;  
evaluate  $P(t)$ ;  
while not terminate do  
     $P'(t) :=$  recombine  $P(t)$ ;  
     $P''(t) :=$  mutate  $P'(t)$ ;  
    evaluate  $P''(t)$ ;  
     $P(t + 1) :=$  select ( $P''(t) \cup Q$ );  
     $t := t + 1$ ;  
end
```

In questo algoritmo, $P(t)$ denota una popolazione di μ individui nell'istante t . Q è un insieme speciale di individui considerati più promettenti per l'operazione di selezione *Es*. $Q = P(t)$; tale selezione è effettuata mediante il calcolo della funzione di fitness che rappresenta il criterio con cui discriminare gli individui. A partire dagli individui selezionati sono effettuate le operazioni che consentono di fare evolvere la popolazione, o in altre parole, di navigare lo spazio di ricerca e di avvicinarsi verso la soluzione ottima del problema. Una prima operazione è la ricombinazione, che consente a partire da due individui di crearne uno nuovo attraverso la combinazione del patrimonio genetico dei due individui di partenza. Una seconda operazione è la mutazione che introduce variabilità nella riproduzione provocando cambiamenti casuali all'interno del patrimonio genetico dell'individuo. Una popolazione discendente $P''(t)$ di grandezza $\lambda \geq \mu$ è generata, quindi, per mezzo di (ricombinazioni e)

mutazioni della popolazione $P(t)$, gli individui discendenti sono valutati calcolando il valore della funzione obiettivo $f(\vec{x}_k)$ per ogni soluzione \vec{x}_k rappresentata dagli individui in $P'(t)$, e la selezione è effettuata mediante il calcolo della funzione di fitness che rappresenta il criterio con cui discriminare tra loro gli individui.

3.2 Applicazione delle tecniche sul caso di studio

Nell'ottimizzazione dello scheduling delle attività di manutenzione elettrica sono stati utilizzati gli algoritmi evolutivi multi obiettivo (Multi Objective Evolutionary Algorithm - MOEA), in particolare sono stati messi a confronto NSGA (**N**on-dominated **S**orting **G**enetic **A**lgorithm) e NSGA II con i risultati ottenuti implementando il modello matematico.

L'NSGA è basato su regole di classificazione degli individui molto rigorose, in quanto, prima della selezione la popolazione viene classificata secondo il concetto di dominanza ovvero tutti gli individui che non sono dominati vengono classificati in gruppi o classi. Allo scopo di mantenere la diversità delle soluzioni, gli individui classificati vengono suddivisi (shared) in base al valore che le funzioni di fitness assumono in essi e, una volta formato il primo gruppo, esso viene momentaneamente ignorato per procedere alla creazione di altri gruppi non dominati di ordine inferiore. Il processo continua fino a quando non vengono classificati tutti gli individui.

Questo algoritmo è stato criticato per tre motivi:

- esso presenta un'alta complessità computazionale, dovuta al processo di ordinamento della popolazione;
- dispendioso quando si usa un alto numero di individui;
- manca di elitismo.

L'NSGA II è un discendente dell'NSGA e, una volta trovati tutti gli elementi non dominati di una popolazione in una determinata generazione, essi vengono assegnati ad un gruppo o rank e tali elementi vengono momentaneamente trascurati per fare in modo che lo stesso procedimento venga applicato agli elementi rimanenti. Alla fine del processo ogni elemento apparterrà ad un gruppo.

La complessità di questa procedura è la somma delle complessità richieste nell'identificazione di ogni insieme non dominato poichè una volta che il primo insieme non dominato è stato individuato, il numero di soluzioni rimanenti è minore del numero originale degli individui della popolazione, e quindi le classificazioni successive richiederanno una complessità computazionale minore.

L'NSGA II fa uso dell'elitismo ovvero gli individui o anche chiamati cromosomi migliori di una generazione saranno presenti anche nella generazione successiva. Per fare in modo che questo accada è necessario che tutti gli individui vengano ordinati, di generazione in generazione, secondo il concetto di dominanza di pareto.

3.2.1 Confronto tra tecniche di risoluzione e risultati ottenuti applicando il modello matematico

I dati per i test sono stati forniti dalla più grande azienda che opera nel settore elettrico in Italia e le istanze prese in esame sono state 3:

- Istanza A: 25 interventi con 4 squadre;
- Istanza B: 30 interventi con 5 squadre;
- Istanza C: 40 interventi con 7 squadre.

Per operare il confronto, si sono valutati:

- il tempo di esecuzione;

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
627	150	80155
553	240	79460
448	120	87291
434	870	75810
424	900	76161
233	420	82002
223	420	82062
-16	480	79460
-21	1110	75977
-66	1500	73758

Tabella 3.1: Risultati ottenuti applicando la tecnica *NSGA* sull'Istanza **A**.

- la qualità delle soluzioni, nelle tre diverse componenti (costo dello straordinario, tempo di interruzione e costo di spostamento).

Nella tabella vengono riportati, per ogni tecnica sperimentata e per ogni istanza del problema, i valori delle prime dieci soluzioni della PFO (Pareto Optimal Front).

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
6658	1380	70455
2201	90	86982
511	660	78390
-663	1770	72822
-1228	1980	70365
-1399	120	84868
-1470	1860	72822
-1553	2310	70365
-1595	180	84015
-1692	900	74410

Tabella 3.2: Risultati ottenuti applicando la tecnica *NSGA* sull'Istanza **B**.

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
1692	690	140139
564	720	140701
-649	810	129998
-1863	870	124838
-1930	960	120098
-1935	960	121111
-2629	960	122126
-2773	1050	121147
-2911	1080	118927
-2959	2790	102866

Tabella 3.3: Risultati ottenuti applicando la tecnica *NSGA* sull'Istanza **C**.

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
383	870	79354
347	420	82104
136	1800	76352
91	240	85811
55	330	82932
51	240	89600
-88	300	89041
-97	240	90415
-148	330	85811
-231	180	94354

Tabella 3.4: Risultati ottenuti applicando la tecnica *NSGA II* sull'Istanza **A**.

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
643	150	78948
527	510	78921
524	870	75258
294	510	79035
281	90	92588
211	120	92166
67	2010	70276
-188	2280	68738
-231	1530	72455
-368	2310	68738

Tabella 3.5: Risultati ottenuti applicando la tecnica *NSGA II* sull'Istanza **B**.

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
-1583	750	142059
-1724	1740	111665
-1729	810	142189
-1768	1680	112265
-1965	870	138398
-2210	840	142217
-2220	2010	110243
-2346	1890	110260
-2557	1770	111665
-2675	1140	130721

Tabella 3.6: Risultati ottenuti applicando la tecnica *NSGA II* sull'Istanza **C**.

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
91	140	79811
55	230	76932
51	140	83600
-16	380	73460
-21	1010	69977
-66	1400	67758
-88	200	83041
-107	140	84415
-158	230	79811
-241	80	88354

Tabella 3.7: Risultati ottenuti utilizzando il software *CPLEX* per risolvere il modello matematico sull'**Istanza A**.

- Il grafico in Fig. 3.1 mostra un confronto sulla base dei tempi di esecuzione dei due algoritmi:
- I grafici che seguono valutano la qualità delle soluzioni ottenute per ogni istanza:

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
511	130	68390
294	90	67035
281	640	62588
-560	120	62166
-1328	90	60276
-1450	100	58738
-1550	1530	59455
-1368	2310	55738
-1663	170	60222
-1228	800	57738

Tabella 3.8: Risultati ottenuti utilizzando il software *CPLEX* per risolvere il modello matematico sull'**Istanza B**.

Costo straordinario $\gamma = 0.34$ (secondi)	Tempo interruzione $\alpha = 0.33$ (secondi)	Costo spostamento $\beta = 0.33$ (metri)
564	600	120701
-1880	770	109998
-1583	750	122059
-1908	820	91665
-2100	810	122189
-2300	800	92265
-2900	870	104838
-2950	960	100098
-2935	960	101111
-3050	870	118398

Tabella 3.9: Risultati ottenuti utilizzando il software *CPLEX* per risolvere il modello matematico sull'**Istanza C**.



Figura 3.1: Confronto dei tempi di esecuzione delle metauristiche *NSGA* e *NSGAII* con il tempo ottenuto dal software *CPLEX*

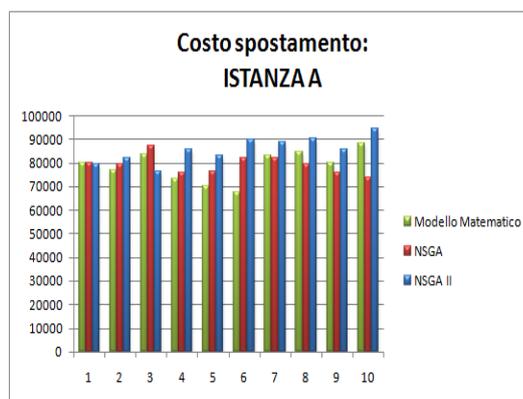


Figura 3.2: Confronto tecniche sull'*Istanza A* in base al parametro *Costo di Spostamento*

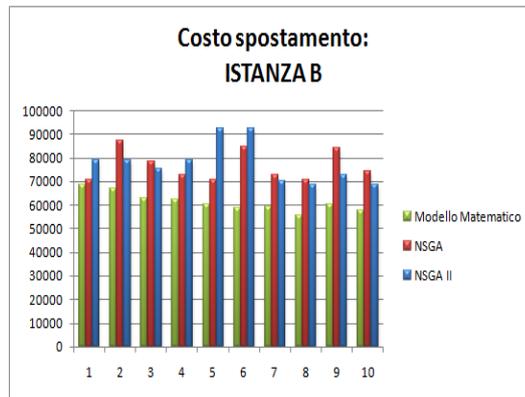


Figura 3.3: Confronto tecniche sull'Istanza B in base al parametro *Costo di Spostamento*

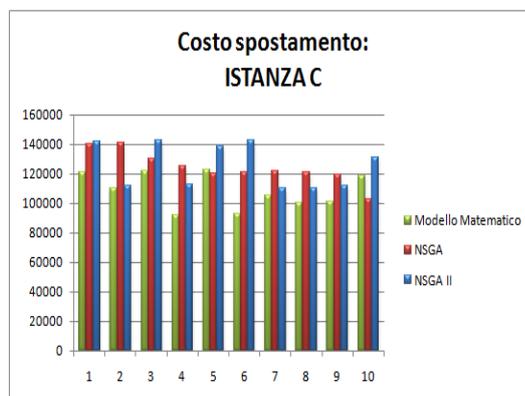


Figura 3.4: Confronto tecniche sull'Istanza C in base al parametro *Costo di Spostamento*

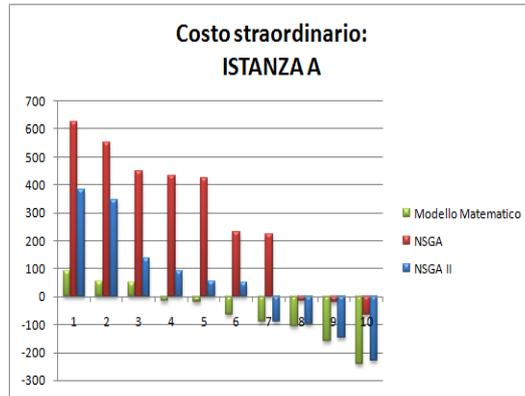


Figura 3.5: Confronto tecniche sull'Istanza A in base al parametro *Costo di Straordinario*

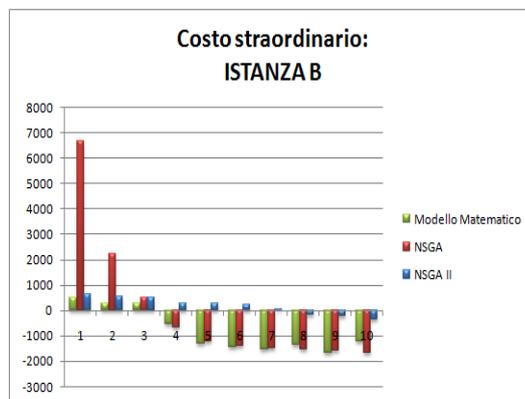


Figura 3.6: Confronto tecniche sull'Istanza B in base al parametro *Costo di Straordinario*

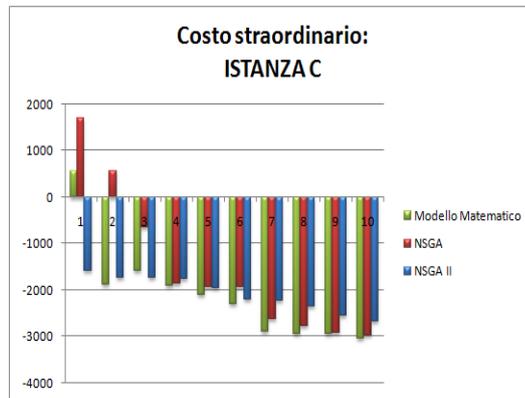


Figura 3.7: Confronto tecniche sull'Istanza C in base al parametro Costo di Straordinario

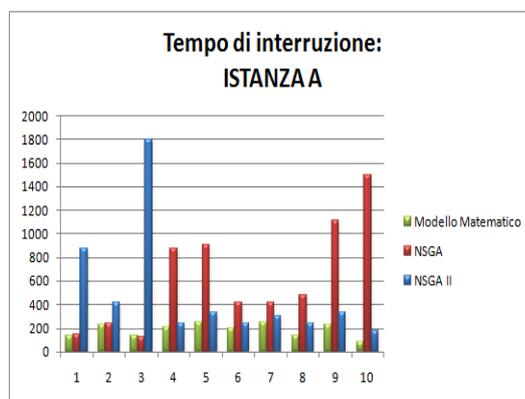


Figura 3.8: Confronto tecniche sull'Istanza A in base al parametro Tempo di interruzione

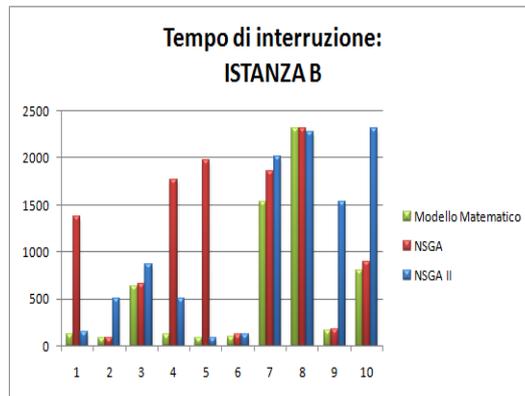


Figura 3.9: Confronto tecniche sull'Istanza B in base al parametro *Tempo di interruzione*

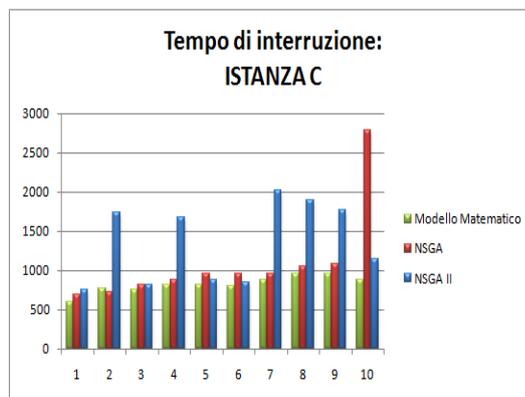


Figura 3.10: Confronto tecniche sull'Istanza C in base al parametro *Tempo di interruzione*

Le tecniche sperimentate, basate su algoritmi genetici, hanno riportato dei risultati interessanti, sia in termini di qualità delle soluzioni che in termini di prestazioni.

Dal punto di vista della qualità delle soluzioni (nelle tre diverse componenti: costo dello straordinario, tempo di interruzione e costo di spostamento) e dei tempi di elaborazione, l'algoritmo NSGA2 ha generato soluzioni con un costo complessivo più basso rispetto a quelle generate dall'algoritmo NSGA.

Conclusioni

L'obiettivo del presente progetto di tesi è lo studio di una soluzione per il problema di assegnazione dinamica di lavori di manutenzione delle reti elettriche.

Il problema può essere così sintetizzato: dato un insieme di lavori, anche chiamati task di manutenzione, che devono essere effettuati da un determinato gruppo di risorse bisogna calcolare sia una giusta allocazione delle risorse ai task che una pianificazione degli interventi in un determinato arco temporale in base a specifici vincoli. I più influenti sono:

- vincoli in materia di lavoro del personale (numero di ore lavorate in una giornata lavorativa, straordinari, ecc.);
- vincoli inerenti le competenze che devono essere possedute dalle risorse per risolvere un determinato intervento;
- vincoli sulle priorità dell'intervento e materiali necessari.

Nello specifico, l'obiettivo è lo sviluppo di un algoritmo di allocazione ottima delle risorse che massimizzi il numero degli interventi della forza lavoro, rispetta i vincoli sopracitati e al contempo garantisca buoni livelli di servizio minimizzando i costi come chilometraggio e straordinario.

Il suddetto problema è di tipo multi-obiettivo e racchiude simultaneamente questioni afferenti a tre classi di problemi: *personnel scheduling problem* (associazione di turni e working days alla forza lavoro), *task assignment problem* (associazione ottima di n agenti su m task, in modo tale che ad ogni task sia puntualmente associato un agente) e *vehicle routing problem* (ottimizzazione degli istradamenti).

Dopo un attento studio dei modelli presenti in letteratura che risolvono problemi analoghi, ma applicati a contesti differenti come ospedali (turnazioni del personale) ed ambiente aeroportuale (*crew rostering*), è stato formulato un modello di schedulazione, basato sulla ricerca operativa e descritto dettagliatamente nel *Capitolo 2*.

Il modello matematico, essendo multi-obiettivo, tende ad ottimizzare diverse funzioni di costo:

- massimizzare la produttività (task schedulati all'interno della finestra temporale predefinita);
- minimizzare i costi operativi (costi del personale e costi degli spostamenti);
- massimizzare la qualità del servizio, in termini di assegnazione ai task di manutenzione della risorsa operativa adeguata.

Tali funzioni obiettivo sono assoggettati a tutta una serie di vincoli atti a descrivere il contesto operativo del problema stesso: priorità, competenze, assegnazione degli interventi, vincoli geografici, ecc.

La verifica funzionale del modello è stata realizzata mediante il noto software di ottimizzazione *IBM ILOG CPLEX* utilizzando scenari reali forniti dalla più grande azienda che opera nel settore elettrico in Italia.

L'elaborazione di una soluzione per il suddetto problema può richiedere tempi di calcolo talmente elevati da essere incompatibile con le necessità operative delle aziende che si occupano di manutenzione delle reti elettriche. Questo perchè il problema affrontato appartiene alla classe dei problemi NP-hard, e quindi difficili da risolvere.

A causa di tale limite, l'attenzione si è focalizzata sull'applicazione di tecniche di ottimizzazione meta-euristiche multi-obiettivo, che pur non garantendo il calcolo della soluzione ottima, sono in grado di determinare in tempi ragionevoli una soluzione spesso vicina all'ottimo globale. In particolare sono state utilizzate le tecniche basate su *Evolutionary Computation* (EC): *NSGA* ed *NSGA2*.

I risultati delle elaborazioni di test sono riportati nella parte finale del *Capitolo 3*. Entrambe le tecniche di risoluzione hanno generato risultati interessanti sia in termini di qualità delle soluzioni che in termini di prestazioni. Tuttavia, l'algoritmo *NSGA2* è risultato il più idoneo per l'implementazione del solver, in quanto ha individuato soluzioni di costo più basso ed in tempi più brevi.

Bibliografia

- [1] D.E. Goldberg, *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, ISBN: 0201157675, 1989.
- [2] J. H. Holland, *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, MI, 1975.
- [3] J. H. Holland, *Outline for a logical theory of adaptive systems*. Journal of the Association for Computing Machinery, Vol. 9 Issue 3, pp. 297-314, 1962.
- [4] L. J. Fogel, *Toward inductive inference automata*, in Proceedings of the International Federation for Information Processing, pp. 395-399, Munich, 1962.
- [5] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, Wiley, New York, 1966.
- [6] G. Dantzing, *A comment on Edie's traffic delay at toll booths*, Operations Research 2, pp. 339-341, 1954.
- [7] G. Weil, K. Heus, P. Francois, M. Poujade, *Constraint programming for nurse scheduling*, IEEE Engineering in Medicine and Biology 14 Vol.4, pp. 417-422, ISSN 0739-5175, 1995.

- [8] A. T. Ernst, H. Jiang, M. Krishnamoorthy, D. Sier, *Staff scheduling and rostering: A review of applications, methods and models*, European Journal of Operational Research 153 pp. 3-27, 2004.
- [9] P. Brucker, R. Qu, E. Burke, *Personnel Scheduling: Models and Complexity*, European Journal of Operational Research, Vol. 210 Issue 3, pp. 467-473, 2011.
- [10] K. A. Downsland, *Nurse scheduling with tabu search and strategic oscillation*, European Journal of Operational Research, Vol. 106, pp. 393-407, 1998.
- [11] L. Pantà, T. Dusan, *Simulated annealing for the multi-objective aircrew rostering problem*, Transportation Research, Part A, Vol. 33, pp 19-45, 1999.
- [12] R. Moore, J. Evans, H. Noo, *Computerized tailored blocking*, in Proceedings of the AGIFORS Symposium, vol. 18, pp. 343–361, 1978.
- [13] J. Byrne, *A preferential bidding system for technical aircrew*, in Proceedings of the AGIFORS Symposium, vol. 28, pp. 87–99, 1988.
- [14] S. Kirkpatrick, L. Gellat, M. Vecchi, *Optimization by Simulated Annealing*, Science, Vol. 220, pp. 671-680, 1983.
- [15] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, *Equation of state calculations by fast computing machines*, Journal of Chemical Physics, Vol. 21 Issue 6, pp. 1087-1092, 1953.
- [16] R. Cipriano, L. Di Gaspero, A. Dovier, *Hybrid Approaches for Rostering: A Case Study in the Integration of Constraint Programming and Local Search*, Lecture Notes in Computer Science, Vol. 4030, pp 110-123, 2006.