



**UNIVERSITÀ
DEGLI STUDI
DI BRESCIA**

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea
in Ingegneria Informatica

Relazione Finale

**MILP: MODELLI DI PROGRAMMAZIONE LINEARE
MISTA INTERA PER LA SCHEDULAZIONE DEL
PERSONALE**

Relatrice:

Chiar.ma Prof. Renata Mansini

Laureando:

Lublanis Matteo

Matricola n. 736418

Anno Accademico 2024/2025

Indice

Introduzione	2
Personnel Scheduling: definizione e caratteristiche	2
Obiettivo e struttura della tesi	3
1 Descrizione del problema	4
1.1 Introduzione	4
1.2 Settimana lavorativa	4
1.3 Tipologia personale, caratteristiche turnazione	4
1.4 Richieste del caporeparto	5
1.5 Skill del personale	5
1.6 Preferenze del personale	6
1.7 Obiettivi finali	6
2 Stato dell'arte	7
2.1 Caratteristiche del personale	7
2.2 Vincoli e Obiettivi	8
2.3 Metodi Risolutivi e Incertezza	10
2.4 Aree di Applicazione	12
2.5 Rassegna bibliografica aggiuntiva	12
3 Definizione formale	16
3.1 Problema base	16
3.2 Problema esteso	21
3.3 Funzione obiettivo	25
4 Implementazione	26
4.1 Gurobi	26
4.2 Strutture dati, rappresentazione dell'istanza e dati tecnici	26
4.3 Risultati implementativi	30
Possibili estensioni e conclusioni	32
A Valori dei coefficienti degli obiettivi	33

Introduzione

Personnel Scheduling: definizione e caratteristiche

Un problema di Personnel Scheduling consiste nel trovare il modo migliore per attribuire i turni ai lavoratori, in modo tale sia da rispettare una serie di vincoli di servizio e contrattuali e sia da massimizzare le preferenze del personale riducendo comunque al minimo i costi previsti.

Negli ultimi decenni, i problemi di schedulazione del personale sono stati studiati ampiamente. Una delle cause principali dietro questo interesse sta il lato economico: il costo del lavoro rappresenta ad oggi, per molte realtà aziendali, uno degli elementi principali della spesa aziendale e riuscire a minimizzarlo può portare grossi benefici all'azienda. La prima formulazione di questo problema venne proposta negli anni '50 da George Dantzig e William Edie[22], ma ad oggi risulta antiquata in quanto:

- Non esistevano diverse tipologie di contratto, solo contratti full-time.
- Ogni dipendente è interscambiabile, non vengono tenute in considerazione le skill del singolo individuo.
- Come obiettivo vi era solo quello di minimizzare il costo del personale, non considerando altri fattori come preferenze o indisposizioni.

Baker[22] fornì una classificazione dei problemi di Personnel Scheduling in tre gruppi:

- Shift Scheduling: si pianifica su un orizzonte di pianificazione giornaliero, implicando che il fabbisogno di personale per ciascun turno può essere gestito in modo indipendente per determinare le allocazioni appropriate; rappresenta il problema più facile da risolvere delle tre categorie, ma non risponde a situazioni più complesse (esempio, la necessità dei turni che si sovrappongono, tipico problema per i call center).
- Days off Scheduling: al posto di pianificare i turni di lavoro, si pianificano i giorni di riposo per i dipendenti; problema tipico per aziende che lavorano sette giorni su sette, mentre il lavoratore ne lavora solo cinque/sei (una variante del problema potrebbe richiedere che i giorni di riposo siano consecutivi).
- Tour scheduling: una combinazione delle due categorie viste precedentemente, bisogna assegnare un tour (giorni della settimana e ore del giorno) da lavorare ad un particolare dipendente. Questa categoria risulta molto più complicata rispetto alle due precedenti e la complessità del problema, seppur dipenda da più fattori, è fortemente influenzata dalla durata minima dell'intervallo di pianificazione (solitamente, dai 15 minuti alle 8 ore).

In Ricerca Operativa, il problema di Personnel Scheduling di tipo NP-hard: la complessità deriva principalmente da tutte le possibili combinazioni dei turni che si possono ottenere, rendendo la ricerca di un ottimo molto più complessa e lunga.

Il problema è in continua evoluzione e necessita quindi modelli complessi e flessibili a eventuali cambiamenti. Gli strumenti risolutivi oggi disponibili (Gurobi, CPLEX, FICO Xpress, Google OR-Tools) integrano algoritmi sempre più raffinati: un problema che decenni fa veniva approcciato manualmente ottenendo risultati approssimativi, può ora essere risolto con precisione, garantendo soluzioni di alta qualità o, in alcuni casi, ottime a livello globale.

Obiettivo e struttura della tesi

Questa tesi si pone come obiettivo quello di risolvere il problema di personnel scheduling (tour scheduling) per il reparto di cassa di un supermercato. Il fine ultimo è quello di trovare la miglior pianificazione settimanale possibile per suddetto supermercato e sviluppare uno strumento applicativo software per permettere un'assegnazione dei turni semplificata.

Nel Capitolo 1 viene presentato informalmente il problema reale considerato, descrivendone i soggetti coinvolti, le preferenze espresse, ciò che si vuole ottenere e i vincoli di varia natura.

Nel Capitolo 2 viene delineato lo stato dell'arte nel settore, andando anche a descrivere altre eventuali applicazioni degli stessi concetti che si tratteranno.

Nel Capitolo 3 si fornisce la descrizione formale del problema, che verrà tradotta in un vero e proprio modello matematico.

Nel Capitolo 4 si illustra lo strumento utilizzato per l'implementazione del modello matematico (Gurobi), le principali fasi del processo di realizzazione e i risultati dei test sperimentali volti a valutarne l'efficienza e la validità.

Negli appendici infine vengono riportati, per ordine:

- Una guida sintetica per l'utilizzo del programma finale, accompagnata da alcuni frammenti di codice commentati.
- Valori dei coefficienti di penalità assegnati ai termini della funzione obiettivo.

1 Descrizione del problema

1.1 Introduzione

La formulazione di un piano orario per i propri dipendenti risulta fattibile fin quando la dimensione del personale da gestire rimane contenuta e fin quando non si tengono in considerazione variabili secondarie come preferenze del personale e limiti giuridici.

Si considera il problema di turnazione per un roster di cassieri di un ipermercato, appartenente alla grande distribuzione organizzata. Il piano di turnazione viene stilato settimanalmente dal caporeparto delle casse e può richiedere fino a due/tre giorni di lavoro intenso, spesso non riuscendo a trovare una soluzione adeguata al flusso della clientela e alle esigenze del punto vendita, creando inoltre disagio all'interno del personale.

Si richiede di formulare un piano di turnazione standard e poi di risolvere il problema di assegnamento dei dipendenti ai corrispettivi turni, sia rispettando i vincoli contrattuali e sia andando incontro a esigenze di varia natura (aziendali e dei singoli dipendenti), in modo tale da non solo ridurre i costi legati alla pianificazione dei turni ma anche massimizzare la soddisfazione del personale.

1.2 Settimana lavorativa

L'ipermercato opera 7 giorni su 7, con un orario continuato per la clientela dalle 08:00 fino alle 20:00. Tralasciando le festività nazionali (Natale, Ferragosto, Festa dei lavoratori, ...), il punto vendita è aperto tutto l'anno.

L'orario per un cassiere normale va dalle 08:00 fino alle 20:30, mezz'ora dopo la chiusura dell'ipermercato, in modo tale da poter eseguire le attività finali necessarie, come conteggio dei contanti e pulizie. Alle 07:00 è richiesto invece l'utilizzo della spazzatrice per poter pulire le corsie: solo alcuni dipendenti possiedono la formazione necessaria per poter adoperare la spazzatrice, dipendenti ai quali potrà essere richiesto di iniziare il turno prima rispetto agli altri.

1.3 Tipologia personale, caratteristiche turnazione

Il personale è suddiviso in tre categorie contrattuali:

- Full-time: 40 ore settimanali;
- Part-time: 24 ore settimanali;
- Part-time weekend: 20 ore settimanali, con disponibilità limitata ai giorni di venerdì (solo turno di chiusura), di sabato e domenica.

Ogni singolo turno può durare dalle 3 alle 5 ore. Non sono previste pause in un turno, ma sono previsti due turni giornalieri per i cassieri full-time e part-time fine settimana

(solo il sabato e la domenica per loro); quindi, ci sarà uno stacco tra due turni giornalieri, che può variare da una mezz'ora fino a quattro ore e mezza. Ai cassieri part-time viene assegnato un solo turno giornaliero. Un dipendente può lavorare fino a 12 ore al giorno, ma è una situazione indesiderata: si tende ad assegnare fino ad un massimo di 8 ore al giorno ad ogni singolo dipendente, con piccole variazioni di mezz'ora in positivo o in negativo in base alle esigenze. Ad un dipendente si possono assegnare fino ad un massimo di 48 ore settimanali (ore previste da contratto più straordinari), ma si vuole rimanere il più possibile vicini ai monte ore previsti da contratto; questo vale anche per i part-time, seppure per essi le ore lavorate oltre le 20/24 sono considerate supplementari (è prevista una maggiorazione minore rispetto agli straordinari), fin quando non si raggiungono le 40.

Tra l'ultimo turno del giorno precedente e il primo turno del giorno preso in considerazione, vi deve essere un riposo di almeno 11 ore previsto per legge. Considerando la struttura degli orari (senza considerare chi deve passare la spazzatrice), si rispetta tranquillamente il vincolo, però in ogni caso risulta preferibile una pausa tra due turni in giorni consecutivi la più lunga possibile, così da permettere un riposo adeguato al dipendente.

1.4 Richieste del caporeparto

Il caporeparto è il soggetto che predispone il fabbisogno per ogni singolo slot temporale, in modo tale da poter pianificare il personale rispondendo in maniera adeguata al flusso della clientela e alle esigenze del punto vendita. Il caporeparto fornisce una tabella settimanale, di cui ogni singola cella rappresenta il numero di dipendenti richiesto per un certo slot orario di uno specifico giorno: ad esempio, il sabato, essendo il giorno più intenso per l'ipermercato, avrà un fabbisogno molto più elevato rispetto ad un giorno infrasettimanale come il martedì. Al momento, non viene fatta una compilazione automatica sulla base dei dati delle vendite e dei documenti dell'ufficio.

Il fabbisogno è un numero indicativo di quante persone siano necessarie in quel momento; non è un limite obbligatorio, quindi possono esserci situazioni di sotto o sovracopertura. La situazione di sovracopertura è preferibile rispetto a quella di sottocopertura, in particolare nel caso in cui vi siano dipendenti che possano andare in scatolame durante situazioni di basso flusso della clientela: una situazione di eccesso di personale in cassa non solo risulterebbe inutile, ma verrebbe anche malvisto dai clienti.

Il caporeparto vuole riuscire a coprire il fabbisogno richiesto, limitando il più possibile il monte ore lavorato totale, così da limitare le spese del personale.

1.5 Skill del personale

Skill particolari individuate durante la raccolta delle informazioni del problema sono:

- Cassa-Reparto: alcuni cassieri possono essere assegnati sia nel reparto scatolame che alle casse, differenziandoli così dai puri cassieri.
- Formazione spazzatrice: alcuni cassieri possiedono la formazione necessaria per adoperare la spazzatrice; tali dipendenti possono iniziare a lavorare prima delle 08:00 per poter così pulire l'ipermercato entro l'apertura.

1.6 Preferenze del personale

Oltre ai vincoli contrattuali, si vuole tenere conto di alcune preferenze espresse dai dipendenti, che contribuiscono a migliorare la soddisfazione e l'equità complessiva della pianificazione. In particolare, vengono considerate le seguenti preferenze:

- che vi sia equità nella distribuzione del carico lavorativo;
- avere giorni di riposo consecutivi;
- ridurre al minimo gli stacchi tra due turni nello stesso giorno (favorendo turni continuativi);
- disporre del riposo domenicale, ove possibile.

1.7 Obiettivi finali

La formulazione del piano dei turni mira al raggiungimento di più obiettivi:

- Minimizzare i costi del personale;
- Massimizzare la soddisfazione del personale;
- Mantenere flessibilità operativa, in modo da poter gestire eventuali imprevisti (assenze, malattie, permessi, variazioni della domanda).

Il problema affrontato è un problema di ottimizzazione combinatoria tipico del rostering/staff scheduling, il quale richiede di assegnare, per un periodo prefissato (una/due settimane, un mese o addirittura un anno), un insieme di turni ai cassieri disponibili, in modo da soddisfare i vincoli di copertura, orario e riposo, cercando contemporaneamente di ottimizzare più criteri di tipo economico e organizzativo.

2 Stato dell'arte

Nel 2012, l'European Journal of Operational Research pubblicò un articolo chiamato *Personnel scheduling: A literature review*[16], nel quale un team di sei ricercatori ha raccolto 291 articoli per dare un quadro generale sullo stato dell'arte per il problema di Personnel Scheduling dal 2004 fino ad allora, definendone inoltre una tassonomia. La tassonomia per i problemi di Personnel Scheduling si basa su quattro campi: caratteristiche del personale, vincoli assieme a misure di performance e flessibilità, metodi risolutivi e incorporazione dell'incertezza, area applicativa e applicabilità della ricerca.

2.1 Caratteristiche del personale

La prima caratteristica del personale che si definisce è la tipologia di contratto: la maggior parte delle pubblicazioni scientifiche riguardante questo problema trattano problemi in cui il personale è per intero di tipo full-time. L'articolo di Hojati e Patil [15] è uno dei pochi che va a trattare la categoria dei contratti part-time per intero. Nel caso dei contratti part-time, la gestione dei turni diventa molto più articolata, in quanto bisogna tenere in considerazione nuovi problemi, come la disponibilità dei dipendenti rispetto a giorni specifici o anche la richiesta necessaria per l'azienda.

In alcuni problemi, le attività possono richiedere alcune skill specifiche, rendendo così il personale considerato ancora più eterogeneo rispetto alla suddivisione "full-time/part-time"; si considerano anche i casi in cui tali attività possono essere svolte da persone che non possiedono le skill necessarie, comportando però un aumento del costo, in quanto saranno ovviamente meno efficienti rispetto al personale qualificato. Le skill specifiche sono in stretto legame con i livelli di produttività e anzianità: allo stesso modo della classificazione in base alle skill, un lavoratore meno produttivo equivarrà ad un costo aumentato. Sia la produttività che l'anzianità possono essere combinate assieme, anche con altre skill. In particolare, per l'anzianità possono essere previsti dei privilegi, come giorni di riposo consecutivi o un peso maggiore che viene dato alle loro preferenze per gli orari.

Un'altra classificazione si basa sul raggruppamento degli impiegati: questo è molto utile in problemi dove si considera lo scheduling di un team di persone e non ogni dipendente individualmente, come ad esempio problemi legati all'area dei trasporti, dove si lega il problema di Personnel Scheduling con quello del routing di veicoli. Esempi per questo tipo di problemi sono l'articolo di Sydney C.K. Chu[6] o anche l'articolo di Heil, Hoffmann e Buscher[23], che si preoccupa di inquadrare nel dettaglio i problemi di crew scheduling per il trasporto ferroviario. Nell'articolo viene anche definita una tassonomia sulla base del tipo di decisioni che devono essere prese (assegnamento delle attività, sequenza dei turni, tempo, altro) e vengono separati in base a se vengono prese per interi team o per singoli membri del personale. Un esempio che evidenzia la tassonomia basata sulle decisioni si ritrova nell'articolo di Horn et al.[7], che si occupa di trovare il modo

più efficiente per la Forza di pattugliamento della Royal Australian Navy di utilizzare una nuova generazione di barche militari. Osservando la letteratura, si può notare che la maggior parte degli articoli si occupa solamente di creare una turnazione accettabile o una schedulazione dei lavori per i dipendenti, dato un carico di lavoro deterministico. Quasi mai, questo problema viene integrato con altri problemi di pianificazione, come la pianificazione del personale, previsione e adeguamento della distribuzione del carico di lavoro, distribuzione delle pause, assunzione e licenziamento, formazione, . . . Diventa importante questo per la ricerca futura: cercare di unire tutte le decisioni in un unico problema di scheduling.

Anche la flessibilità con il tempo ha assunto un ruolo fondamentale, rendendo ancora di più complicata la formulazione dei modelli per i problemi considerati. La flessibilità può riguardare: sovrapponibilità dei turni, diversi orari di inizio, diverse lunghezze dei turni, diverse durate delle pause.

Le scelte decisionali prese per lo scheduling diventano fondamentali non solo per questioni economiche, ma anche per la questione psicologica e fisica dei lavoratori. Xu e Hall[24] pubblicarono un articolo riguardante la fatica lavorativa, introducendola come variabile all'interno dei problemi di scheduling: un lavoratore stanco rende di meno, aumentando di conseguenza il costo complessivo. L'articolo va ad analizzare quanto in letteratura venga misurata e analizzata la fatica e quale sia il suo impatto sulla performance, andando ad analizzare l'impatto che scelte di scheduling possono avere su di essa. Questo tema si lega strettamente al concetto di flessibilità, in quanto uno schema rigido non può rispondere ad una situazione variabile come la fatica di un lavoratore e bisognerà introdurre nuovi concetti, come quello delle rate-modifying activity (RMA, azioni che cambiano la velocità di produzione di una macchina) o delle micro-pause.

2.2 Vincoli e Obiettivi

Si differenziano i vincoli in hard e soft, ove con hard si indicano vincoli che non possono essere violati, mentre soft sono vincoli più flessibili. Nell'articolo vengono definite le seguenti categorie per i vincoli: copertura, temporali, equità e di equilibrio.

Sia i vincoli hard che i vincoli soft di copertura stanno alla base dell'intero problema di scheduling, ovvero la scelta del numero di dipendenti necessari per coprire un carico di lavoro. I vincoli di copertura vanno contro solitamente la funzione obiettivo, che cerca di minimizzare la forza lavoro totale, in quanto direttamente legata al costo. La differenza tra capacità del personale minima e ottima è di fondamentale interesse per chi si occupa di questi problemi, in quanto rende necessario affrontare problemi come l'eccesso o il difetto di personale. All'interno dell'articolo, vengono riportati alcuni documenti che vanno anche a trattare problemi che vanno al di là della semplice gestione dei turni del personale, come la carenza di personale a livello internazionale per problemi di scheduling di infermieri: una possibile soluzione è quella di definire i vincoli di copertura come soft, prevedendo l'introduzione di personale esterno per coprire buchi all'interno

della pianificazione e tenendo come obiettivo quello di generare un insieme di roster che minimizzino il numero di turni scoperti durante il lasso temporale considerato. L'utilizzo di vincoli hard implica quasi sempre, all'interno della letteratura trovata, un divieto per l'understaffing: nell'articolo, viene citato un singolo caso in cui la situazione è invertita ed è dovuto a motivi finanziari (sempre legati alla mancanza di personale). Altra cosa che si potrebbe implementare nella copertura riguarda le pause, però essendo un problema di tipo principalmente real-time non vengono trattate all'interno delle pubblicazioni sui problemi di personnel scheduling (Thompson e Pullman han dedicato un intero articolo soltanto alla gestione delle pause[9], andando ad analizzarne i pro e i contro di una organizzazione delle pause anticipata).

Se il personale può essere caratterizzato da delle skill, si introducono solitamente vincoli hard per assicurarsi la presenza di dipendenti qualificati per lo svolgimento di un particolare lavoro. I vincoli soft in questo caso possono venire utilizzati per indicare una possibile introduzione di personale non adatto all'attività considerata, implicando però un aumento del costo, o anche possono rappresentare il concetto di skill alternative, ovvero dipendenti che possiedono diverse skill ma che preferiscono evitare di coprire certi tipi di lavoro. Si distinguono tre diversi gruppi in base al grado di flessibilità delle competenze:

- Le skill sono definibili a livello di dipendente, ovvero chi pianifica ha piena libertà di scegliere le skill per ogni membro del personale.
- L'insieme dei dipendenti viene gerarchizzato: dipendenti a livelli superiori possono eseguire attività dei dipendenti a livelli inferiori, mentre non vale il contrario; fattori su cui si può basare la gerarchia sono esperienza, dipendenti junior/senior, background formativo.
- Le skill non possono essere sostituite, alcune attività richiedono skill specifiche e possono essere svolte solo da lavoratori con quelle particolari skill.

Altro problema importante è quello degli straordinari, il cui utilizzo può influenzare molto sulla flessibilità della copertura. Gli straordinari possono essere limitati settimanalmente, oppure possono essere studiati su orizzonti temporali più ampi, lasciando più libertà di gestione degli orari settimanali: un esempio nel quale lo studio delle ore complessive viene fatto mensilmente è il caso studiato nella tesi di Serena Cortopassi[17], dove viene studiato un problema di personnel scheduling per il personale de Il Gignoro, una residenza assistenziale per anziani.

Le misure finanziarie prevedono costi differenti, come costo del personale, possibile costo relativo ad un particolare giorno della settimana, diversi costi per diverse skill, costo degli straordinari, costo legato allo svolgimento di particolari attività: minimizzare il costo è strettamente legato al minimizzare il numero di dipendenti usati, ma questo non significa che è solo questo ma è anzi un problema molto più complesso che va

analizzato. Già modellare il costo del problema attorno al costo del personale invece che il numero dei dipendenti, si riesce ad ottenere un compromesso tra assunzione di dipendenti, straordinari, lavoratori occasionali e altro. Altri possibili costi possono essere costi di produzione, costi per chiamate perse o mancata produzione, costi differenti tra lavoratori full-time e part-time, costi di rifiuto, costi differenti per sede.

Spesso gli operatori vogliono anche garantire equità nella gestione degli orari per i dipendenti, che si può tradurre in più punti: distribuzione equa di weekend di riposo, un numero di turni sfavorevole bilanciato tra i vari dipendenti, gestione bilanciata per tipo di turno, quantità bilanciata di giorni di lavoro e di riposo (viene citato il lavoro di Lezaun et al.[8]), gestione delle preferenze dei dipendenti (ad esempio, preferenze per certi tipi di turni o anche volere lavorare con specifici collaboratori), numero di giorni di riposo consecutivi).

2.3 Metodi Risolutivi e Incertezza

In letteratura, si trova un insieme veramente ampio e diversificato di tecniche risolutive adottate. Si dividono le tecniche usate in tre gruppi:

- Programmazione matematica: programmazione intera, programmazione dinamica, goal programming, euristiche costruttive o migliorative.
- Simulazione, programmazione a vincoli, queuing.
- Altro: soluzioni meno frequenti come approssimazione lineare a tratti di una curva, analytic hierarchy process (AHP), modelli di fogli di calcolo e data envelopment analysis (DEA).

Il primo gruppo rappresenta il gruppo più ampio: il problema del personnel scheduling viene modellato come un programma lineare, intero o misto intero e la formulazione più usata è quella del Set Covering, introdotta da Dantzig. Il problema del *Set Covering* consiste nel trovare la più piccola sotto-collezione di insiemi (sottoinsiemi dell'insieme universo) tale per cui l'unione dei suoi elementi sia equivalente all'insieme universo:

$$\begin{aligned} \min \quad & \sum_{s \in S} x_s \\ \text{s.t.} \quad & \sum_{s: e \in s} x_s \geq 1, \forall e \in U \\ & x_s \in \{0, 1\} \end{aligned}$$

In particolare, nel caso del personnel scheduling, al posto dell'1 nel secondo vincolo si avrà il fabbisogno necessario per lo slot temporale considerato. Questa formulazione è molto comoda, perché permette di aggiungere a proprio piacimento vincoli aggiuntivi in base alle proprie necessità.

Una tecnica che aiuta molto, in particolare per problemi di larga scala, è quella della decomposizione del problema: viene separato il problema in due parti, una più facile e una più complessa e per ognuna posso scegliere il metodo risolutivo da utilizzare. Diversi modi di applicare la decomposizione furono proposti da: Detienne et al.[12] sfruttarono la decomposizione di Bender (problema principale rilassato è un problema di zaino multidimensionale a multiscelta, mentre ogni sotto problema è un problema di b-matching¹, Bard e Wan[10] [5]. Assieme alla decomposizione, un'altra tecnica buona per problemi di grandi dimensioni è quella della generazione di colonne: quando le variabili in gioco sono troppe, si rischia di eccedere il limite di memoria assegnato per il problema, quindi vengono generate dinamicamente quando necessario[32] (vengono citati per questo approccio Beliën e Demeulemeester[11]). Tramite la decomposizione, si riesce ad utilizzare a rappresentare il problema con altri modelli oltre a quello del set covering, come ad esempio problemi di maximum flow.

Ultimo approccio molto utilizzato all'interno della programmazione matematica è quello delle metaeuristiche, che sono progettate in modo tale da trovare soluzioni ammissibili buone in tempi relativamente brevi o accettabili dai limiti imposti. Gli svantaggi delle metaeuristiche sta nel fatto che non possono in modo dimostrabile né produrre soluzioni ottime né ridurre lo spazio di ricerca delle soluzioni. Gli algoritmi preferiti sono la tabu search e gli algoritmi genetici all'uso di algoritmi di ricottura simulata. Altre alternative sono: scatter search, ricerca locale iterata, variable neighborhood search, particle swarm optimization, algoritmi memetici, reti neurali, . . .

Trattando invece l'incertezza, si differenziano tre tipi di incertezze che si possono ritrovare all'interno di questi problemi:

- Incertezza della richiesta: carico del lavoro imprevedibile.
- Incertezza dell'arrivo: pattern imprevedibile dell'arrivo carico di lavoro (fallimenti della macchina, arrivo di chiamate per i call center).
- Incertezza della capacità: deviazione tra forza lavoro pianificata e effettiva.

Si dividono inoltre gli approcci in due tipi:

- Deterministico: viene ignorata ogni forma di incertezza.
- Stocastico: viene considerata l'incertezza (non per forza tutti i tipi visti prima).

L'approccio deterministico è quello dominante. Non incorporare l'incertezza all'interno del modello non equivale però a considerare il carico di lavoro costante in ogni momento: vengono fatte analisi del carico di lavoro osservando i dati storici e facendone una stima.

¹Generalizzazione del problema base di matching per un grafo. Sia dato un grafo $G = (V, E)$, ad ogni vertice $v \in V$ viene assegnato una capacità $b(v) \in \mathbb{N}^+$: un b-matching è un sottoinsieme di archi tale che, per ogni v , il numero di archi incidenti a v non superi la sua capacità $b(v)$.

Per andare contro la variabilità del carico di lavoro, negli approcci deterministici può risultare d'aiuto l'utilizzo di buffer di capacità in modo tale da rendere il roster del personale molto più robusto.

Cause di incertezza relative ai dipendenti possono essere: malattie dei dipendenti, ritardi negli arrivi, perdita di capacità per via di risorse fuori servizio. L'incertezza di arrivo è di principale interesse per i sistemi di call center, i quali spesso devono aggiungere anche l'incertezza di richiesta.

2.4 Aree di Applicazione

Si distinguono sei aree di applicazione: servizi, trasporti, generale, manifattura, vendita al dettaglio (retail) e problemi di scheduling per il personale militare. Il problema più trattato riguarda quello della gestione dei turni degli infermieri e rappresenta una delle fondamenta in tutta la letteratura riguardante i problemi di scheduling.

2.5 Rassegna bibliografica aggiuntiva

Un articolo di Brucker et al.[14], pubblicato nel 2011, sempre dall'European Journal of Operational Research, va a studiare più dettagliatamente i modelli proposti e la complessità relativa per i problemi di Personnel Scheduling. Nell'articolo si rifanno alla classificazione delle aziende, proposta da De Causmaecker, in quattro categorie in base al tipo di problema riscontrato: pianificazione centrata sulla permanenza, pianificazione centrata sulla fluttuazione, pianificazione centrata sulla mobilità e pianificazione centrata sul progetto. Un problema di scheduling per il personale di un ospedale si differenzia molto rispetto a problemi di scheduling per il settore dei trasporti. Il problema che viene affrontato all'interno dell'articolo è quello di cercare di trovare un modello matematico generale che possa portare con sé caratteristiche comuni nei diversi problemi di scheduling del personale. Viene fornito un modello matematico generale per il problema studiato, che viene riportato, tralasciando l'ultima parte sui vincoli per la richiesta flessibile e il limite superiore per il numero di dipendenti assegnati:

”Vi è un orizzonte di pianificazione $[0, T]$ diviso in periodi $[t, t+1[$ con $t = 0, 1, \dots, T-1$. Durante l'orizzonte di pianificazione vi sono m compiti da eseguire ($j = 1, \dots, m$). $D_j(t)$ è il numero di dipendenti necessari per eseguire il compito j nel periodo $[t, t+1[$. Viene chiamato il profilo di domanda (fabbisogno).

Si ha un insieme E di n dipendenti. Ad ogni $e \in E$ si ha un sottoinsieme Q_e di compiti per cui e è qualificato e può quindi essere assegnato; e non può essere assegnato a compiti al di fuori di Q_e . Un pattern di lavoro per un dipendente e si definisce con:

- un vettore zero-uno $(w_e(t))_{t=0}^{T-1}$ dove $w_e(t)$ vale 1 se e solo se e è disponibile nel periodo $[t, t+1[$ e

- l'assegnazione di compiti appartenenti a Q_e per ogni periodo di tempo $[t, t+1[$ con $w_e(t) = 1$.

I pattern di lavoro sono rappresentata da vettori binari $\pi = (\pi(j, t))$ ove $\pi(j, t) = 1$ se e solo se $[t, t+1[$ è un periodo di lavoro nel quale il compito j deve essere svolto.

Si definisce P_e l'insieme di tutti i pattern lavorativi ammissibili per il dipendente e . Bisogna assegnare ai dipendenti $e \in E$ dei pattern lavorativi ammissibili tali per cui venga coperta la richiesta $D_j(t)$ per ogni task j e tali per cui i costi relativi alle assegnazioni ai dipendenti sia minimizzato.

Si possono imporre anche "vincoli soft" nei pattern lavorativi e aggiungere una penalità $u(\pi_e)$ che va a definire di quanto un pattern π_e vada a violare il vincolo soft. Possibili costi di assegnazione possono essere: la somma delle penalità di tutti i dipendenti assegnati, il numero di dipendenti richiesti, i costi per i dipendenti assegnati. Il modello può essere più o meno restrittivo in base al problema considerato: si può introdurre il concetto di richiesta flessibile, ove ogni compito j ha una durata p_j e deve essere eseguito entro una finestra temporale $[L_j, R_j[\supset [0, T]$ con $R_j - L_j \geq p_j$, ovvero $D_j \in [0, 1]$ per $t = L_j, L_j + 1, \dots, R_j - 1$ tale che $\sum_{t=L_j}^{R_j-1} D_j(t) = p_j$.

Il problema può essere modellato come un problema lineare binario. Sia $x_{e\pi}$ una variabile binaria che vale 1 solo e soltanto se il pattern lavorativo π viene assegnato al dipendente e . In aggiunta, sia $c_{e\pi}$ i costi dovuti all'assegnamento del pattern π al lavoratore e . La formulazione sarà:

$$\begin{aligned}
& \min \sum_{e \in E} \sum_{\pi \in P_e} c_{e\pi} x_{e\pi} \\
& \text{subject to} \\
& \sum_{\pi \in P_e} x_{e\pi} \leq 1, e \in E \\
& \sum_{e \in E} \sum_{\pi \in P_e} \pi(j, t) x_{e\pi} \geq D_j(t), \text{ for all } (j, t) \\
& x_{e\pi} \in \{0, 1\} \text{ for all } e \in E \text{ and } \pi \in P_e
\end{aligned}$$

Viene assegnato al più un pattern π al dipendente e . In alcuni casi, bisogna assegnare ad ogni dipendente un pattern lavorativo, cambiando il primo vincolo da disuguaglianza a uguaglianza pari ad uno."

Nel caso speciale per cui la richiesta per ogni compito j è costante e ogni dipendente è disponibile in tutti i periodi, questo modello può essere formulato come un problema del flusso di costo minimo² e può essere quindi risolto in tempi polinomiali. Tuttavia,

²Il problema del flusso di costo minimo (minimum-cost flow problem, abbreviato MCFP) è un problema di decisione e ottimizzazione che consiste nel trovare il modo meno costoso possibile di far passare un certo ammontare di flusso tramite una rete di flusso.

nella realtà queste condizioni non vi sono quasi mai, quindi di norma un problema di scheduling appartiene alla classe dei problemi di tipo NP-Complete. Si è trovato anche che l'utilizzo di algoritmi di flusso di rete può essere utile per risolvere problemi di assegnamento dei turni ai dipendenti e dei dipendenti ai diversi compiti: questo può essere sfruttato all'interno delle euristiche utilizzate per lo scheduling del personale.

Un altro articolo, "*Polynomially solvable personnel rostering problems*" [19], va a studiare i sottoproblemi che stanno alla base dei problemi di personnel scheduling, andando a separarli in problemi facili e difficili (in base al tempo risolutivo richiesto). Mediante l'utilizzo di trasformazioni di problemi differenti in problemi di flusso di costo minimo, è stato possibile identificare nuovi casi, rispetto ai quattro citati nell'introduzione, che possono essere risolti in tempi polinomiali. Per i vincoli di successione e di counter (numero di giorni lavorati e numero di turni lavorati per tipo), si è trovato che la differenza tra problemi semplici e difficili sta nella definizione dei vincoli a livello di giorno o a livello di turno. Conoscendo questo, è possibile non solo riformulare il problema (riportandolo a problemi già conosciuti, come i problemi di flusso), in quanto potrebbe rendere il problema trattabile computazionalmente, ma si può anche offrire approcci efficienti verso la risoluzione dei sottoproblemi che nascono dalla decomposizione del problema.

In letteratura, sono riuscito a trovare una sola pubblicazione che parla direttamente del problema affrontato in questa tesi ed è un articolo di Melachrinoudis e Olafsson [1] del 1994. In questo articolo, viene fornito un modello di programmazione lineare intera per lo scheduling dei cassieri. Il modello viene eseguito in un ambiente basato su fogli di calcolo, rendendolo agevole per i dirigenti. Vengono usati tre pacchetti software: fogli di calcolo Lotus 1-2-3 in combinazione con "What's Best", un pacchetto di ottimizzazione sviluppato da LINDO.

- Un pacchetto di previsione del lavoro POS (collezionatore di dati che veniva eseguito sul computer mainframe del supermercato). Vengono solitamente salvati i dati relativi alle ultime quattro settimane e possono essere alterati tenendo in considerazione fattori secondari, come periodi festivi. Vengono definiti, sulla base di questi dati, un valore di massimo per i clienti in attesa alla cassa e un valore su come il carico di lavoro influenzi la produttività dei cassieri.
- Per rappresentare i dati, vengono usati i fogli di calcolo Lotus 1-2-3, in quanto agevoli all'utilizzo per i direttori dei negozi. Questo pacchetto fa da collegamento tra le altre due parti del sistema ed è strutturato in quattro sezioni: sezione dei cassieri schedulati (si usano variabili decisionali per rappresentare il numero di cassieri schedulati nel turno di lunghezza j che inizia al tempo k), sezione dei vincoli orari dei cassieri (viene usata dal dirigente, cambia le condizioni per cui un cassiere viene assegnato ad un turno), sezione del personale di cassa (contenente il fabbisogno per slot temporale) e sezione del totale (rappresenta la somma degli elementi delle tre colonne nella sezione del personale di cassa).

- "What's Best", un pacchetto di ottimizzazione sviluppato da LINDO.

Il modello fornito a What's Best si può trovare nell'articolo citato, si tiene precisare che la formulazione si basa sul problema di Set Covering.

Infine, si riporta l'articolo di Agnetis et al.[27], ove viene descritto lo stato dell'arte per i problemi di scheduling per macchinari. I concetti riportati, per quanto distanti dal problema analizzato in questa tesi, possono essere comunque d'aiuto per affrontare il problema di personnel scheduling.

3 Definizione formale

In questa sezione, si offre una descrizione formale del problema mediante l'utilizzo di un modello di Programmazione Lineare Mista Intera (MILP), che verrà poi inserito come input in Gurobi, un risolutore matematico. Un modello si struttura nel seguente modo:

$$\begin{aligned} \min / \max_{x \in \mathbb{Z}^n} \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \geq 0 \end{aligned}$$

ove $\min / \max_{x \in \mathbb{Z}^n} c^T x$ rappresenta la funzione obiettivo che deve essere massimizzata o minimizzata, mentre le altre due righe rappresentano i vincoli del modello. Questa sezione è suddivisa in due parti:

1. Problema base: viene definito formalmente il minimo necessario per avere un modello base funzionante, senza andare incontro a vincoli aggiuntivi come skill o preferenze del personale.
2. Problema esteso: vengono aggiunti i vincoli mancanti al problema base, analizzando anche quali elementi potrebbero andare in conflitto con quelli precedenti.

3.1 Problema base

Si definisce l'insieme C come l'insieme dei cassieri. Al momento, i cassieri si differenziano solo per il monte orario previsto da contratto:

$$C = FT \cup PT \cup PTW \quad (3.1)$$

Appartenere a categorie differenti equivale soltanto ad avere monte orari differenti: per il momento, tutti i cassieri condividono la stessa paga oraria (non vi sono differenze di anzianità o di livello contrattuale). Un cassiere c può essere quindi definito nel seguente modo:

$$c(\text{info}, \text{variabili}) \quad (3.2)$$

ove *info* rappresenta i dati del cassiere (non utili per la risoluzione, come nome e cognome) e *variabili* che sono gli effettivi dati utili al problema (monte ore previsto, numero massimo di turni assegnabili giornalieri, skill, etc.).

Durante una giornata, l'obiettivo del caporeparto è quello di coprire il flusso della clientela in maniera adeguata. Rifacendosi al problema del *Set Covering*, all'articolo di Brucker[14] e a quello di Gusmeroli e Bettinelli[26], si definiscono:

$$\begin{aligned} D &= \{1, \dots, m\} \text{ i giorni della finestra temporale.} \\ f(d, k) &\text{ il fabbisogno per lo slot } k \text{ del giorno } d. \\ T(d) &\text{ l'insieme degli slot temporali del giorno } d. \end{aligned}$$

Nel caso studiato, m vale 7. Gli slot temporali nei problemi di scheduling possono avere durata variabile, dall'intera giornata lavorativa fino a soli 15 minuti: si è deciso di utilizzare uno slot temporale di mezz'ora. La giornata di lavoro inizia alle 08:00 e finisce alle 20:30 (verrà modificato non appena si introdurranno i vincoli per la spazzatrice), è quindi suddivisa in 25 slot temporali. L'orario è continuato ed è identico per tutti e sette i giorni della settimana, dunque vi sarà da definire un totale di 175 fabbisogni, ognuno per il singolo slot k della giornata d . Il fabbisogno viene rappresentato mediante l'utilizzo di una matrice, dove i giorni rappresentano le righe e gli slot le colonne. I turni assegnati ai dipendenti saranno gli insiemi tipici del *Set Covering*. Si specificano ora i turni:

$$s(day, start, end, dur) \in S$$

S rappresenta l'insieme dei turni s , i quali sono caratterizzati da:

- day: giorno del turno.
- start: slot d'inizio del turno.
- end: slot di fine del turno (inclusivo, indica l'ultimo slot lavorato).
- dur: durata del turno espressa in quantitativo di slot.

Si definisce anche l'insieme $S(d) \subset S$, ovvero l'insieme dei turni del giorno d . Si definisce anche per i turni il valore:

$$gap(s1, s2) \in \mathbb{Z}_{\geq 0} \quad (3.3)$$

che rappresenta, per ogni coppia di turni $s1$ e $s2$ con $day(s1) = day(s2)$, lo stacco tra i due turni espressa in slot. $gap(s1, s2)$ vale:

$$gap(s1, s2) = \begin{cases} 0 & \text{se } overlap(s1, s2) = 1 \\ start(s2) - end(s1) & \text{se } day(s1) = day(s2) \wedge end(s1) \leq start(s2) \\ start(s1) - end(s2) & \text{se } day(s1) = day(s2) \wedge end(s2) \leq start(s1) \end{cases} \quad (3.4)$$

Per indicare se due turni si sovrappongono, si specifica:

$$overlap(s1, s2) \in \{0, 1\} \quad (3.5)$$

$$overlap(s1, s2) = 1 \iff day(s1) = day(s2) \wedge [start(s1) \leq end(s2) \wedge start(s2) \leq end(s1)] \quad (3.6)$$

che vale 1 se i turni $s1$ e $s2$ sono sovrapposti. La durata dei turni varia dalle 3 alle 5 ore, quindi dai 6 ai 10 slot temporali. Gli slot temporali dovranno essere coperti da almeno $f_{d,k}$ turni. Per dire che un turno s copre uno slot k del giorno d , si usa:

$$copre(s, d, k) \in \{0, 1\} \quad (3.7)$$

Per definire se un cassiere lavora o meno un certo turno, si usa una variabile decisionale:

$$x(c, s) \in \{0, 1\}$$

la quale varrà 1 nel caso in cui al cassiere c sia assegnato il turno s , 0 altrimenti.

Si definisce la variabile:

$$H_{settimana}(c) = 0.5 * \sum_{s \in S} dur(s) * x(c, s) \quad (3.8)$$

$$H_{settimana}(c) \in \mathbb{R} \quad (3.9)$$

che rappresenta le ore totali assegnate al cassiere c . Si moltiplica per 0.5 in quanto due slot rappresentano un'ora.

Ogni cassiere dovrà lavorare il monte orario previsto fino ad un massimale da stabilire:

$$H_{settimana}(c) \geq Mh(c) \quad \forall c \quad (3.10)$$

$$H_{settimana}(c) \leq maxOre(c) \quad \forall c \quad (3.11)$$

$maxOre(c)$ può essere definito diversamente per ogni singolo cassiere oppure essere costante per tutti. Per adesso, viene considerato costante e pari a 48, come descritto nella sezione 1.

I cassieri part-time sono gli unici ai quali viene assegnato un singolo turno al giorno, mentre ai full-time e ai part-time fine settimana possono essere assegnati due turni giornalieri:

Part time weekend: 20 ore settimanali, con disponibilità limitata ai giorni di venerdì (solo il turno di chiusura), sabato e domenica...

...ma sono previsti due turni giornalieri per i cassieri full-time e part-time fine settimana (solo il sabato e la domenica per loro)... Ai cassieri part-time viene assegnato un solo turno giornaliero.

Un dipendente può lavorare fino a 12 ore al giorno, ma è una situazione indesiderata: si tende ad assegnare fino ad un massimo di 8 ore al giorno ad ogni singolo dipendente...

Si esprime il relativo vincolo per il numero di turni giornalieri:

$$0 \leq \sum_{S(d)} x(c, s) \leq maxTurni(c, d) \quad \forall d, \forall c \quad (3.12)$$

$$maxTurni(c, d) = \begin{cases} 2 & \text{se } c \in FT \vee (c \in PTW \wedge d \in \{6, 7\}) \\ 1 & \text{se } c \in PT \vee (c \in PTW \wedge d \in \{5\} \wedge k \geq k(16 : 30)) \\ 0 & \text{altrimenti} \end{cases}$$

Per il numero di ore giornaliere assegnate a un cassiere c si definisce:

$$H_{giorno}(c, d) = 0.5 * \sum_{S(d)} x(c, s) dur(s) \quad (3.13)$$

$$H_{giorno}(c, d) \in \mathbb{R} \quad (3.14)$$

Deve valere:

$$H_{giorno}(c, d) \leq 12 \quad \forall c, \forall d \quad (3.15)$$

in quanto ogni cassiere può lavorare fino ad un massimo di 12 ore.

Per assegnare due turni ad un cassiere nello stesso giorno, bisogna assicurarsi che i due turni non si sovrappongano:

$$x(c, s_1) + x(c, s_2) \leq 1 \quad \forall c \in C, \forall d \in D, \forall s_1 < s_2 \in S(d) : overlap(s_1, s_2) = 1 \quad (3.16)$$

La copertura non può essere sempre coperta rigidamente, in quanto il personale può non essere adeguato all'effettiva richiesta in alcune situazioni. Considerato questo, si introducono due nuove variabili:

$$cover(d, k) = \sum_{S(d)} \sum_{c \in C} x(c, s) copre(s, d, k) \quad (3.17)$$

$$u(d, k) \geq 0 \quad (3.18)$$

$$cover(d, k), u(d, k) \in \mathbb{N} \quad (3.19)$$

con le quali si garantisce la copertura mediante il seguente vincolo:

$$cover(d, k) + u(d, k) \geq f(d, k) \quad \forall d \in D, \forall k \in T(d) \quad (3.20)$$

La copertura del fabbisogno rappresenta l'obiettivo principale dell'intero problema, per cui:

$$f_1 = \sum_{d, k} u(d, k) \quad (3.21)$$

andrà minimizzato. Si vuole anche ridurre le ore complessive lavorate dai cassieri:

$$f_2 = \sum_{c \in C} H_{settimana}(c) \quad (3.22)$$

I cassieri hanno richiesto uno stacco tra due turni giornalieri il più breve possibile. Questo permetterebbe di ridurre al minimo il tempo trascorso sul luogo di lavoro e aumentare la soddisfazione del personale. Il problema si pone solo nel caso in cui il cassiere faccia due turni in un giorno. Si definisce la seguente variabile binaria:

$$split(c, d) \in \{0, 1\} \quad \forall c \in FT \cup PTW \quad (3.23)$$

che indica se al lavoratore c vengono assegnati due turni lo stesso giorno d . Deve valere:

$$\sum_{s \in S(d)} x(c, s) \leq 1 + \text{split}(c, d) \quad \forall c \in FT \cup PTW, \forall d \quad (3.24)$$

$$\sum_{s \in S(d)} x(c, s) \geq 2 * \text{split}(c, d) \quad \forall c \in FT \cup PTW, \forall d \quad (3.25)$$

In quanto si è assicurato della non sovrapposizione, possiamo definire:

$$\text{stacco}(c, d) \in \mathbb{N} \quad (3.26)$$

una variabile intera che indica lo stacco tra due turni nel giorno d per il cassiere c espresso in numero di slot temporali. Lo stacco sarà determinato da:

$$\text{stacco}(c, d) \leq \text{maxGap} \cdot \text{split}(c, d) \quad (3.27)$$

$$\begin{aligned} \text{stacco}(c, d) &\geq \text{gap}(s1, s2) * (x(c, s1) + x(c, s2) - 1) \quad \forall c \in FT \cup PTW, \forall d, \\ &\quad \forall (s1, s2) \in S(d), \\ &\quad s1 \neq s2 \wedge s1 < s2 \end{aligned} \quad (3.28)$$

che andrà minimizzato:

$$f_3 = \sum_{d \in D} \sum_{c \in FT \cup PTW} \text{stacco}(c, d) \quad (3.29)$$

Si definisce la variabile:

$$\text{riposo}(c, d) \in \{0, 1\}$$

che vale 1 se e solo se il cassiere c non è stato assegnato a nessun turno il giorno d , 0 altrimenti. Utilizzando un m-vincolo, si definisce la variabile prima spiegata:

$$\sum_{s \in S(d)} x(c, s) \leq \text{maxTurni}(c, d)(1 - \text{riposo}(c, d)) \quad \forall c \in FT \cup PT, \forall d \quad (3.30)$$

$$\sum_{s \in S(d)} x(c, s) \geq 1 - \text{maxTurni}(c, d) * \text{riposo}(c, d) \quad \forall c \in FT \cup PT, \forall d \quad (3.31)$$

Non si studia il problema per i lavoratori part-time fine settimana, in quanto possono essere assegnati il sabato e la domenica.

Durante la settimana, il cassiere deve avere almeno un giorno di riposo e per evitare un sovraccarico di ore giornaliere (avendo per esempio solo 4 giorni lavorativi da 10 ore ciascuno e 3 di riposo) può arrivare ad un massimo di due giorni di riposo settimanali:

$$1 \leq \sum_{d \in D} \text{riposo}(c, d) \leq 2 \quad \forall c \in FT \cup PT \quad (3.32)$$

3.2 Problema esteso

Straordinari Gli straordinari vengono penalizzati in quanto prevedono una retribuzione maggiorata, aumentando notevolmente il costo finale per il punto vendita. Si introduce:

$$O(c) \in \mathbb{R}_{\geq 0}$$

che rappresenta le ore straordinarie assegnate al cassiere c . Questo valore è maggiore o uguale a 0 e si preferirebbe che valesse 0 per tutti i cassieri del roster. Non vale 0 nel caso in cui:

$$O(c) \geq H_{settimana}(c) - Mh(c) \quad \forall c \quad (3.33)$$

ovvero quando al cassiere c vengono assegnate più ore di quelle previste dal contratto. La somma di questi valori va minimizzata:

$$f_4 = \sum_{c \in C} O(c) \quad (3.34)$$

Non si fanno distinzioni tra supplementari e straordinari: sono entrambi penalizzati allo stesso modo. Il peso assegnato a questo obiettivo può variare in base al periodo, rendendo più favorevoli gli straordinari in periodi più critici, come quelli festivi.

Equità I problemi riguardanti l'equità sono:

- Carico orario settimanale uguale per tutti i cassieri dello stesso tipo.
- Preferenza globale sulla durata dei turni.
- Nel caso dei cassieri full-time e dei part-time, la gestione delle domeniche libere rientra nel principio di equità.

Per il primo punto, si definiscono:

$$\bar{H}_{settimana}(t) \in \mathbb{R} \quad \forall t \in type(= FT, PT, PTW) \quad (3.35)$$

$$dev(c) \geq 0 \quad \forall c \in C \quad (3.36)$$

dove $\bar{H}_{settimana}(t)$ rappresenta la media delle ore assegnate ai cassieri del tipo t , mentre $dev(c)$ sta a indicare la deviazione assoluta per singolo cassiere c dalle ore medie assegnate. Varranno i seguenti vincoli:

$$\bar{H}_{settimana}(t) = \frac{1}{|t|} \sum_{c \in t} H_{settimana}(c) \quad \forall t \in type \quad (3.37)$$

$$dev(c) \geq H_{settimana}(c) - \bar{H}_{settimana}(t) \quad \forall t \in type, \forall c \in t \quad (3.38)$$

$$dev(c) \geq \bar{H}_{settimana}(t) - H_{settimana}(c) \quad \forall t \in type, \forall c \in t \quad (3.39)$$

Si dovrà minimizzare la deviazione totale:

$$f_5 = \sum_{c \in C} dev(c) \quad (3.40)$$

Altro problema da considerare è la durata dei turni. Sono permessi turni dalle 3 fino alle 5 ore, ma una durata eccessiva dei turni non è ben accetta dal personale e potrebbe portare a situazioni di squilibrio, in cui alcuni dipendenti lavorano un solo turno da 5 ore al giorno mentre altri ne lavorano due da 3/4 ore ciascuno. Per questione di praticità, possiamo imporre un'unica preferenza per l'intero personale verso il turno da 4 ore (si potrebbe andare a specificare per ogni dipendente una preferenza specifica). Si introduce:

$$pen(s) = \begin{cases} p_3 & \text{se } dur(s) = 6 \\ p_{3.5} & \text{se } dur(s) = 7 \\ 0 & \text{se } dur(s) = 8 \\ p_{4.5} & \text{se } dur(s) = 9 \\ p_5 & \text{se } dur(s) = 10 \end{cases} \quad (3.41)$$

che indica le penalità relative ai tipi di turni separati per durata. Si minimizza quindi:

$$f_6 = \sum_{s \in S, c \in C} pen(s)x(c, s) \quad (3.42)$$

Questa equità non è intesa come confronto diretto tra cassieri, ma come preferenza globale del personale verso determinate durate di turno; l'introduzione di uno standard comune favorisce una distribuzione più omogenea delle tipologie di turno, senza imporre vincoli di equità individuale. Ogni cassiere potrebbe esprimere la propria preferenza singola, rendendo la penalità dipendente da c , ovvero $pen(c, s)$.

I cassieri full-time e part-time hanno espresso la loro preferenza verso il riposo domenicale. La finestra temporale attuale è:

$$D = 1, \dots, m \quad \text{con } m = 7$$

La domenica è identificata da $d = 7$. Possiamo utilizzare di nuovo la variabile $riposo(c, d)$ e massimizzare così il riposo domenicale:

$$f_7 = \sum_{c \in (FT \cup PT)} riposo(c, 7) \quad (3.43)$$

Riposo continuato Si introduce una nuova variabile binaria per indicare il riposo continuato per i cassieri full-time:

$$riposoContinuato(c, d) \in 0, 1$$

che indica se per due giorni consecutivi il lavoratore è stato a riposo. Per fare valere questa definizione, si specifica:

$$riposoContinuato(c, d) \leq riposo(c, d) \quad \forall c \in FT, \forall d \in 1, \dots, m-1 \quad (3.44)$$

$$riposoContinuato(c, d) \leq riposo(c, d+1) \quad \forall c \in FT, \forall d \in 1, \dots, m-1 \quad (3.45)$$

$$riposoContinuato(c, d) \geq riposo(c, d) + riposo(c, d+1) - 1 \quad \forall c \in FT, \forall d \in 1, \dots, m-1 \quad (3.46)$$

$$\sum_{d=1}^{m-1} riposoContinuato(c, d) \leq 1 \quad \forall c \in FT \quad (3.47)$$

Si vuole massimizzare questa quantità in quanto richiesto dal personale:

$$f_8 = \sum_{c \in (FT)} \sum_{d \in 1, \dots, m-1} riposoContinuato(c, d) \quad (3.48)$$

Cassa-reparto I vincoli soft relativi al fabbisogno non sfavoriscono la situazione generale di deviazione dal fabbisogno dichiarato: questo porta a situazioni di sottocopertura sfavorite di molto rispetto a quelle di sovracopertura, che non vengono in alcun modo influenzate dai vincoli. Si introducono:

$$under(d, k) \in \mathbb{R} \geq 0 \quad (\text{carenza}) \quad (3.49)$$

$$over(d, k) \in \mathbb{R} \geq 0 \quad (\text{sovracopertura}) \quad (3.50)$$

dove $under(d, k)$ indica di quanto si è sotto il fabbisogno nello slot (d, k) , mentre $over(d, k)$ indica di quanto si è sopra.

La copertura effettiva dello slot (d, k) è data dalla somma dei turni assegnati che coprono lo slot, pesati in base alla skill cassa-reparto:

$$cop(d, k) = \sum_{c \in C} \sum_{s \in S(d)} pesoCopertura(c) * copre(s, d, k) * x(c, s) \quad (3.51)$$

La variabile introdotta è molto simile a $cover(d, k)$, ma introduce un nuovo valore che è $pesoCopertura(c)$, necessario per esprimere formalmente la skill di cassa-reparto. Un cassiere che possiede la skill cassa-reparto ha un peso minore nel conteggio della sovracopertura, in quanto può uscire in corsia in caso di eccesso di cassieri in cassa, riducendo il carico sul reparto.

Si definisce quindi:

$$pesoCopertura(c) = \begin{cases} \rho_1 & \text{se } c \text{ non possiede la skill cassa-reparto} \\ \rho_2 & \text{se } c \text{ possiede la skill cassa-reparto} \end{cases} \quad (3.52)$$

$$\rho_1 = 1; 0 < \rho_2 < 1$$

La deviazione assoluta dal fabbisogno viene modellata imponendo:

$$\text{cop}(d, k) + U(d, k) - O(d, k) = f(d, k) \quad \forall d \in D, ; k \in T(d) \quad (3.53)$$

In questo modo:

- $\text{cop}(d, k) < f(d, k), U(d, k) > 0$ (carenza);
- $\text{cop}(d, k) > f(d, k), O(d, k) > 0$ (sovracopertura);
- $\text{cop}(d, k) = f(d, k), U(d, k) = O(d, k) = 0$.

L'obiettivo da minimizzare è il seguente:

$$f_9 = \sum_{d \in D} \sum_{k \in T(d)} (U(d, k) + O(d, k)) \quad (3.54)$$

Questa funzione obiettivo può essere divisa in due per riuscire a gestire situazioni di carenza e sovracopertura in maniera differente:

$$f_{9under} = \sum_{d \in D} \sum_{k \in T(d)} U(d, k) \quad (3.55)$$

$$f_{9over} = \sum_{d \in D} \sum_{k \in T(d)} O(d, k) \quad (3.56)$$

Spazzatrice La spazzatrice, per via delle sue dimensioni, può solo essere operata prima dell'apertura dell'ipermercato alla clientela. Il compito viene assegnato principalmente a due cassieri appartenenti al personale interno disponibile, poiché la variabilità del personale addetto alle pulizie, dovuta al fatto che l'ipermercato si affida ad un'azienda esterna, rende più affidabile l'impiego di dipendenti interni. Vengono definiti due nuovi slot per giornata (prima 1=08:00 e 25=20:00, ora 1=07:00 e 27=20:00), che possono essere coperti soltanto dal personale formato. Si introduce così una nuova categoria:

$$c(Mh, info, cassareparto, spazzatrice) \quad (3.57)$$

$$C^{SP} = \{c \in C : c(spazzatrice) = 1\} \quad (3.58)$$

Si introduce una nuova skill, la quale, se non è presente, non permette la copertura di alcuni slot temporali, in quanto prevede attività diverse da quelle viste fino ad adesso:

$$\begin{aligned} x(c, s) = 0 \quad & \forall c \notin C^{SP}, \forall d \in D, \\ & \forall s \in S(d) : (copre(s, d, 1) = 1 \vee copre(s, d, 2) = 1) \end{aligned} \quad (3.59)$$

La copertura del fabbisogno per gli slot 1,2 è garantita dallo stesso vincolo di copertura 3.20. Per via del vincolo del riposo obbligatorio tra due giornate, questi cassieri non possono venire pianificati in chiusura (ultimo slot) se il giorno dopo iniziano alle 07:00. Si pregenera un insieme delle coppie di turni che non possono essere pianificati assieme per un singolo cassiere:

$$I^{11} = \{(s1, s2) : day(s2) = day(s1) + 1 \wedge \text{riposo tra } s1 \text{ e } s2 \text{ inferiore alle 11 ore}\} \quad (3.60)$$

Varrà di conseguenza:

$$x(c, s_1) + x(c, s_2) \leq 1 \quad \forall (s_1, s_2) \in I^{11}, \forall c \in C \quad (3.61)$$

$$(3.62)$$

In quanto la finestra temporale è limitata ad una singola settimana, il problema del controllo del rispetto del vincolo delle 11 ore di riposo tra la domenica e il lunedì può essere spostato dalla definizione del modello al problema dell'assegnamento dei turni.

3.3 Funzione obiettivo

$$\min \sum_{i=1}^9 \lambda_i f_i \quad \lambda_i \in \mathbb{R} \quad (3.63)$$

I λ_i rappresentano i pesi che noi diamo ai vari obiettivi: più λ_i è elevato, più l'obiettivo relativo incide nella funzione obiettivo generale, ottenendo di fatto più importanza nella ricerca dell'ottimo. λ_i può essere negativo, in modo tale da massimizzare il valore del relativo obiettivo.

4 Implementazione

In seguito all'analisi e alla formulazione del modello relativo al problema, si è sviluppato uno strumento software basato su un solver per verificare la validità dei concetti prima esposti. Lo strumento software è stato sviluppato in Java e come solver è stato utilizzato Gurobi. In questa sezione si presenterà suddetto strumento e si esporranno le strutture dati utilizzate.

4.1 Gurobi

Gurobi Optimizer è un solver commerciale sviluppato da Gurobi Optimization, LLC. Il nome viene dagli autori Dr. Zonghao Gu, Dr. Edward Rothberg e Dr. Robert Bixby, che fondarono Gurobi nel 2008. Gurobi viene utilizzato per problemi di programmazione lineare, programmazione quadratica, programmazione a vincoli quadratici, programmazione lineare mista intera, programmazione quadratica mista intera e programmazione a vincoli quadratici mista intera. Ad oggi, Gurobi viene utilizzato da più di 40 industrie ed è uno dei solver più utilizzati dalle aziende big tech[34].

4.2 Strutture dati, rappresentazione dell'istanza e dati tecnici

Si è diviso il progetto in due pacchetti:

- **model:** contiene le classi relative al modello e ai suoi vincoli, assieme ad una classe di util per la configurazione e la classe Main.
- **data:** contiene le classi e le informazioni utili a organizzare i dati necessari per il modello.

All'interno di data, è presente una raccolta di istanze divise in BAx.txt e EXx.txt, in base al tipo di problema che si vuole risolvere (base o esteso). Questi file contengono i dati relativi al fabbisogno e ai lavoratori, dei quali si salvano alcuni dati anagrafici, la tipologia contrattuale e le skill che possiede nel caso degli EXx.txt. La classe Worker serve per salvare i dati relativi al cassiere, di cui:

- **EmploymentType:** un enum che contiene le informazioni relative alla tipologia contrattuale, come monte ore settimanale previsto, ore settimanali massime assegnabili e numero massimo di turni giornalieri.
- **id:** un long che rappresenta univocamente il cassiere.

Le skill vengono salvate mediante un enum chiamato, sorprendentemente, Skill. La classe Instance si occupa della preparazione dei dati necessari per il modello, come i turni e l'insieme dei lavoratori. All'interno di Instance, viene anche pregenerato l'insieme I11, che contiene le coppie di turni. Le strutture dati principali utilizzate sono:

- List: per salvare i turni.
- Set: per salvare i lavoratori, separati per tipologia contrattuale.
- Map: per collegare tra loro diversi dati, come il turno agli slot che copre.

Per il trasferimento di dati, rispetto a turni o anche successivamente ai lavoratori rispetto ai turni, si utilizzano i Record. I dati vengono letti mediante parsing dai file di testo (questo più avanti verrà ripreso come possibile punto di estensione, introducendo file .json o .xml).

Si passa ora al pacchetto model, il quale è separato nelle seguenti classi:

- Config: una classe di configurazione, contiene costanti utili per impostare il problema;
- Main: esegue il loop di ottimizzazione sulle istanze generate e richiama i metodi necessari per stampare i risultati;
- PersonnelSchedulingBaseModel e ExtendedModel: il cuore dell'intero progetto, in quanto contengono tutti i vincoli relativi al modello;
- SolutionPrinter: stampa le soluzioni su un file di .csv. estendere questo?

All'interno di PersonnelSchedulingBaseModel, tramite la programmazione lineare si rappresentano i vincoli; si riporta un esempio di codice che rimanda ai vincoli di massimo numero di turni assegnabili per giorno:

```

1 public void maxTurnsPerDay() throws GRBException {
2     for (Worker w : instance.getAllWorkers()) {
3         for (int day = 1; day <= Instance.TIMEWINDOW; day++) {
4
5             int maxT = w.getType().maxTurns(day);
6             GRBLinExpr lhs = new GRBLinExpr();
7
8             for (Shift s : instance.getDayByShifts(day)) {
9                 GRBVar xcs = x_CS_Var.get(new WorkerShift(w.getId(), s.id()));
10                if (xcs != null) lhs.addTerm(1.0, xcs);
11            }
12
13            model.addConstr(lhs, GRB.LESS_EQUAL, maxT,
14                "MaxTurns_w" + w.getId() + "_d" + day);
15        }
16    }
17 }

```

Listato 1: Esempio di metodo per i vincoli

Le strutture dati utilizzate sono le stesse utilizzate per Instance, con la differenza che in queste due classi le Map assumono un ruolo più fondamentale, in quanto stanno alla base dell'intera costruzione del modello collegando le variabili di Gurobi ai relativi dati. Le Map utilizzate sono le seguenti:

```
1 private final Map<DaySlot, GRBVar> u_DK = new HashMap<>();
2 private final Map<WorkerShift, GRBVar> x_CS_Var = new HashMap<>();
3 private final Map<WorkerDay, GRBVar> splitC_D = new HashMap<>();
4 private final Map<WorkerDay, GRBVar> staccoC_D = new HashMap<>();
5 private final Map<WorkerDay, GRBVar> riposoC_D = new HashMap<>();
```

Listato 2: Map usate per le variabili del modello

DaySlot, *WorkerShift* e *WorkerDay* sono dei Record. Si utilizzano le seguenti classi di Gurobi per implementare il modello:

- GRBEnv: l'environment per Gurobi, fondamentale per impostare tutto.
- GRBModel: l'oggetto che rappresenta il modello per Gurobi. I metodi utilizzati per questa classe sono: addVar e addConstr per aggiungere variabili e vincoli, setObjective per impostare l'obiettivo, optimize per ottimizzarlo.
- GRBVar: le variabili del modello per Gurobi, aggiunte mediante addVar al modello.
- GRBLinExpr: un'espressione lineare, importante per la definizione dei vincoli.

Le funzioni obiettivo sono salvate mediante GRBLinExpr. Per la funzione obiettivo si sono trovate due strade:

- Metodo pesato: come espresso nella sezione 3, si sommano tutti gli obiettivi assegnandogli i specifici pesi e si esegue poi l'ottimizzazione del modello.
- Metodo gerarchico: si dà una priorità agli obiettivi, si ottimizza prima rispetto all'obiettivo con priorità più alta, si passa poi a quelli con priorità più bassa aggiungendo un vincolo che garantisca il non peggioramento dell'obiettivo precedente.

In Gurobi, per entrambi gli approcci si può adoperare il seguente metodo:

```
1 void setObjectiveN(GRBLinExpr expr, int index, int priority, double
    weight, double abstol, double reltol, String name)
```

ove:

- expr: obiettivo.
- index: indice dell'obiettivo, utile per il log.
- priority: priorità all'interno della gerarchia di ottimizzazione.

- **weight**: peso assegnato all'obiettivo.
- **abstol**: tolleranza assoluta per l'obiettivo, ovvero di quanto può degradare la soluzione quando si ottimizzano obiettivi secondari (stessa unità dell'obiettivo).
- **reitol**: tolleranza relativa per l'obiettivo, come abstol ma relativo (percentuale dell'obiettivo).
- **name**: nome dell'obiettivo, utile per il log.

Per esprimere correttamente quanto esposto nella sezione 3, il metodo è stato usato con la priorità imposta uguale per tutti gli obiettivi, tuttavia si potrebbe comunque risolvere il problema dando priorità maggiore alla copertura e a eventuali obiettivi di equità e minimizzazione delle ore, così da garantire una soluzione migliore per gli obiettivi principali.

Si introduce infine la classe `SolutionPrinter`. Questa classe contiene metodi utili alla stampa della soluzione trovata da Gurobi, in particolare risultano fondamentali:

- **writeSolutionCsvTablesByWorker**: salva su un file `.csv` gli orari separati per singolo cassiere.
- **exportUtoCsv**: salva su un file `.csv` una tabella contenente i valori di $u(d, k)$ per ogni slot temporale, mostrando come si comporta Gurobi.
- **writeObjectiveResult**: ritorna un file `.txt` con i valori trovati per singolo obiettivo.

Si riporta un esempio di tabella oraria generata da `writeSolutionCsvTablesByWorker`:

F. Saruzzi				
giorno				
lun	-	-	-	-
mar	-	-	-	-
mer	-	-	-	-
gio	-	-	-	-
ven	-	-	16:30	20:30
sab	8:00	13:00	14:00	19:00
dom	8:00	11:00	12:30	16:30

4.3 Risultati implementativi

In quanto disponibili, gli esperimenti sono stati eseguiti su una macchina con le seguenti specifiche tecniche:

- AMD Ryzen™ 7 5700X CPU @ 3.40GHz \times 16.
- 32GB RAM 3200MHz.

I test sono stati eseguiti su 3 istanze per il modello base e 3 istanze per il modello esteso, dando 3 diversi limiti di tempo (5, 10 e 30 minuti). I valori dei pesi sono riportati nell'appendice, mentre per i relativi significati dietro i valori dei singoli obiettivo si rimanda alla sezione 3.

Le versioni di Java e Gurobi utilizzate sono rispettivamente Java-21 e Gurobi 13.0.0.

BA1		Valore ottimo		1185.5
Tempo limite	300	600	1800	
Valore f.o	3171.5	2985.0	1271.5	
Gap(%)	62.6	60.3	6.76	
f_1	1980.0	1800.0	40.0	
f_2	1098.5	1110.0	1150.5	
f_3	93.0	75.0	81.0	

BA2		Valore ottimo		8608.0
Tempo limite	300	600	1800	
Valore f.o	8624.0	8624.0	8624.0	
Gap(%)	0.19	0.19	0.19	
f_1	8080.0	8080.0	8080.0	
f_2	516.0	516.0	516.0	
f_3	28.0	28.0	28.0	

BA3		Valore ottimo		5206.0
Tempo limite	300	600	1800	
Valore f.o	5242.0	5240.0	5235.0	
Gap(%)	0.69	0.65	0.55	
f_1	4340.0	4340.0	4340.0	
f_2	846.0	846.0	846.0	
f_3	56.0	54.0	49.0	

EX1	Valore ottimo		473.0
Tempo limite	300	600	1800
Valore f.o	474.0	473.5	473.5
Gap(%)	0.21	0.11	0.11
f_1	0.0	0.0	0.0
f_2	336.0	336.0	336.0
f_3	24.0	24.0	24.0
f_4	0.0	0.0	0.0
f_5	0.0	0.0	0.0
f_6	0.0	0.0	0.0
f_7	-2.0	-2.0	-2.0
f_8	0.0	0.0	0.0
f_9	116.0	115.5	115.5

EX2	Valore ottimo		10271.0
Tempo limite	300	600	1800
Valore f.o	10272.5	10272.5	10272.5
Gap(%)	0.01	0.01	0.01
f_1	8220.0	8220.0	8220.0
f_2	516.0	516.0	516.0
f_3	36.0	36.0	36.0
f_4	92.0	92.0	92.0
f_5	0.0	0.0	0.0
f_6	180.0	180.0	180.0
f_7	-4.5	-4.5	-4.5
f_8	0.0	0.0	0.0
f_9	1233.0	1233.0	1233.0

EX3	Valore ottimo		1419.3
Tempo limite	300	600	1800
Valore f.o	1819.7	1693.9	1669.3
Gap(%)	22.0	16.2	15.0
f_1	120.0	0.0	0.0
f_2	1181.0	1194.5	1187.5
f_3	218.0	208.0	205.0
f_4	89.0	102.5	95.5
f_5	21.2	14.4	10.8
f_6	143.0	135.0	138.0
f_7	-5.5	-5.5	-5.5
f_8	0.0	0.0	0.0
f_9	53.0	45.0	38.0

Possibili estensioni e conclusioni

Capitolo dedicato a possibili estensioni e conclusioni sui dati. Prima di questo bisogna introdurre i grafici, per le estensioni possibili idee sono:

- Shift design
- Basi di dati
- Ampliare finestra temporale
- Euristiche
- Grafica
- Client-Server

A Valori dei coefficienti degli obiettivi

λ_1	20.0
λ_2	1.0
λ_3	1.0
λ_4	1.0
λ_5	1.0
λ_6	1.0
λ_7	-0.5
λ_8	-1.0
λ_{9over}	3.0
λ_{9under}	0.5

Riferimenti bibliografici

- [1] Emanuel Melachrinoudis e Michael Olafsson. «A scheduling system for supermarket cashiers». In: *Computers and Industrial Engineering* 23.1 (1992), pp. 121–124. DOI: [https://doi.org/10.1016/0360-8352\(92\)90078-X](https://doi.org/10.1016/0360-8352(92)90078-X). URL: <https://www.sciencedirect.com/science/article/pii/036083529290078X>.
- [2] Michael J. Brusco e Larry W. Jacobs. «Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations». In: *European Journal of Operational Research* 86.2 (1995), pp. 249–261. DOI: [https://doi.org/10.1016/0377-2217\(94\)00063-I](https://doi.org/10.1016/0377-2217(94)00063-I). URL: <https://www.sciencedirect.com/science/article/pii/037722179400063I>.
- [3] Hesham K. Alfares. «Survey, categorization, and comparison of recent tour scheduling literature». In: *Annals of Operations Research* 127.1 (2004), pp. 145–175.
- [4] Andreas T. Ernst et al. «An annotated bibliography of personnel scheduling and rostering». In: *Annals of Operations Research* 127.1 (2004), pp. 21–144.
- [5] Jonathan F Bard e Lin Wan. «The task assignment problem for unrestricted movement between workstation groups». In: *Journal of Scheduling* 9.4 (2006), pp. 315–341.
- [6] Sydney C.K. Chu. «Generating, scheduling and rostering of shift crew-duties: Applications at the Hong Kong International Airport». In: *European Journal of Operational Research* 177.3 (2007), pp. 1764–1778. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2005.10.008>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221705006430>.
- [7] M E T Horn, H Jiang e P Kilby. «Scheduling patrol boats and crews for the Royal Australian Navy». In: *Journal of the Operational Research Society* 58.10 (2007), pp. 1284–1293. DOI: [10.1057/palgrave.jors.2602300](https://doi.org/10.1057/palgrave.jors.2602300). URL: <https://doi.org/10.1057/palgrave.jors.2602300>.
- [8] Mikel Lezaun, Gloria Pérez e Eduardo Sáinz De La Maza. «Rostering in a rail passenger carrier». In: *Journal of Scheduling* 10.4 (2007), pp. 245–254.
- [9] Gary M. Thompson e Madeleine E. Pullman. «Scheduling workforce relief breaks in advance versus in real-time». In: *European Journal of Operational Research* 181.1 (2007), pp. 139–155. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2006.05.018>. URL: <https://www.sciencedirect.com/science/article/pii/S037722170600381X>.
- [10] Jonathan F Bard e Lin Wan. «Workforce design with movement restrictions between workstation groups». In: *Manufacturing & Service Operations Management* 10.1 (2008), pp. 24–42.

- [11] Jeroen Beliën e Erik Demeulemeester. «A branch-and-price approach for integrating nurse and surgery scheduling». In: *European journal of operational research* 189.3 (2008), pp. 652–668.
- [12] Boris Detienne et al. «Cut generation for an employee timetabling problem». In: *European Journal of Operational Research* 197.3 (2009), pp. 1178–1184.
- [13] Andreas Beer et al. «An AI-Based Break-Scheduling System for Supervisory Personnel». In: *IEEE Intelligent Systems* 25.2 (2010).
- [14] Peter Brucker, Rong Qu e Edmund Burke. «Personnel scheduling: Models and complexity». In: *European Journal of Operational Research* 210.3 (2011), pp. 467–473. DOI: <https://doi.org/10.1016/j.ejor.2010.11.017>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221710007897>.
- [15] Mehran Hojati e Ashok S. Patil. «An integer linear programming-based heuristic for scheduling heterogeneous, part-time service employees». In: *European Journal of Operational Research* 209.1 (2011), pp. 37–50. DOI: <https://doi.org/10.1016/j.ejor.2010.09.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221710006004>.
- [16] Jorne Van den Bergh et al. «Personnel scheduling: A literature review». In: *European Journal of Operational Research* 226.3 (2013), pp. 367–385. DOI: <https://doi.org/10.1016/j.ejor.2012.11.029>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221712008776>.
- [17] Serena Cortopassi. «Modello di Ottimizzazione per la Schedulazione del Personale Infermieristico di una Residenza Assistita». Tesi di laurea triennale in Matematica. Università di Pisa, 2014. URL: <https://etd.adm.unipi.it/t/etd-04172014-181428/>.
- [18] Dingding Lin et al. «Scheduling workforce for retail stores with employee preferences». In: *2015 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. 2015, pp. 37–42. DOI: 10.1109/SOLI.2015.7367407.
- [19] Pieter Smet et al. «Polynomially solvable personnel rostering problems». In: *European Journal of Operational Research* 249.1 (2016), pp. 67–75. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2015.08.025>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221715007626>.
- [20] Michael Mac-Vicar et al. «Real-time recovering strategies on personnel scheduling in the retail industry». In: *Computers and Industrial Engineering* 113 (2017), pp. 589–601. DOI: <https://doi.org/10.1016/j.cie.2017.09.045>. URL: <https://www.sciencedirect.com/science/article/pii/S0360835217304618>.
- [21] Melanie Erhard et al. «State of the art in physician scheduling». In: *European Journal of Operational Research* 265 (2018). URL: <https://www.sciencedirect.com/science/article/pii/S0377221717305787>.

- [22] Shandong Mou, David J. Robb e Nicole DeHoratius. «Retail store operations: Literature review and research directions». In: *European Journal of Operational Research* 265 (2018). DOI: <https://doi.org/10.1016/j.ejor.2017.07.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221717306306>.
- [23] Julia Heil, Kirsten Hoffmann e Udo Buscher. «Railway crew scheduling: Models, methods and applications». In: *European Journal of Operational Research* 283 (2020). DOI: <https://doi.org/10.1016/j.ejor.2019.06.016>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221719304916>.
- [24] Shuling Xu e Nicholas G. Hall. «Fatigue, personnel scheduling and operations: Review and research opportunities». In: *European Journal of Operational Research* 295 (2021). DOI: <https://doi.org/10.1016/j.ejor.2021.03.036>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221721002629>.
- [25] Renato Bruni. *Problemi di Personnel Scheduling*. Materiale didattico, Corso di Ottimizzazione Combinatoria. 2023. URL: https://www.diag.uniroma1.it/~bruni/files/OC11_Scheduling.pdf.
- [26] Nicolo Gusmeroli e Andrea Bettinelli. *A Mixed-Integer Linear Program to create the shifts in a supermarket*. 2024. arXiv: 2403.17850 [math.OC]. URL: <https://arxiv.org/abs/2403.17850>.
- [27] Jeroen Beliën et al. «Fifty years of operational research applied to healthcare». In: *European Journal of Operational Research* 326.2 (2025). DOI: <https://doi.org/10.1016/j.ejor.2024.12.040>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221724009822>.
- [28] Tin-Chih Toly Chen. «Artificial Intelligence (AI) Applications in Job Sequencing and Scheduling». In: *Explainable and Customizable Job Sequencing and Scheduling*. Springer Nature Switzerland, 2025. URL: https://doi.org/10.1007/978-3-031-85374-6_2.
- [29] Müfide Narlı e Onur Derse. «Optimal Crew Scheduling in an Intensive Care Unit: A Case Study in a University Hospital». In: *Applied Sciences* 15.7 (2025). URL: <https://www.mdpi.com/2076-3417/15/7/3610>.
- [30] Zilu Wang, Zhixing Luo e Huaxiao Shen. «Data-driven robust flexible personnel scheduling». In: *Computers & Operations Research* 176 (2025). DOI: <https://doi.org/10.1016/j.cor.2024.106935>. URL: <https://www.sciencedirect.com/science/article/pii/S0305054824004076>.
- [31] Alexey Bochkarev et al. «Quantum computing for discrete optimization: A highlight of three technologies». In: *European Journal of Operational Research* 329.3 (2026), pp. 747–766. DOI: <https://doi.org/10.1016/j.ejor.2025.07.063>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221725005880>.

- [32] *Che cos'è la generazione di colonne?* URL: <https://www.ibm.com/docs/it/icos/22.1.2?topic=example-what-is-column-generation>.
- [33] *Gurobi 13.0 documentation.* URL: <https://docs.gurobi.com/current/>.
- [34] *Gurobi Optimization.* URL: <https://www.gurobi.com>.

Ringraziamenti

Desidero innanzitutto ringraziare la relatrice di questa tesi, Renata Mansini, per l'attenzione e la disponibilità dimostrate durante la stesura.

Ringrazio poi i miei nonni e mia sorella Isabel per il supporto mostrato nel corso di questi ultimi anni.

Ringrazio inoltre le persone che ho incontrato e che mi hanno aiutato durante questo percorso: “maritozzo fan club”, “Il gruppo di Yousef”, “Gli amici di Pietro”, “Il gruppo di Uovo”, Marco, Andrea e il gruppo di lavoro.

Infine, un ringraziamento particolare va a Francesco Sarubbi e Francesco Cominelli, senza i quali questa tesi non sarebbe stata possibile.