**SHIFT-SCHEDULING per CLINICHE e CASE DI CURA**

**Formulazione di vincoli, variabili e funzione obiettivo**

**Michele Grieco**

## **Indice**

### **Inquadramento e impostazione progettuale**

* Obiettivo
* Caratteristiche principali
* Scelte di modellazione

### **Definizione dei vincoli**

* Vincoli di copertura della domanda
* Vincoli orari
* Vincoli sui turni notturni
* Vincoli sui pattern
* Vincoli di riposo
* Vincoli sul saldo ore progressivo
* Vincolo di stabilità

### **Funzione obiettivo e modellazione matematica**

* Struttura generale
* Componenti principali della funzione obiettivo  
   a. Slack dei vincoli soft  
   b. Saldo ore progressive  
   c. Penalità per turni in un altro reparto  
   d. Fairness su notti, festivi e weekend

e. Stabilità del piano

f. Funzione obiettivo complessiva

g. Possibili sviluppi e integrazioni

1. **Valutazione delle soluzioni e gestione delle criticità**

* Valutazione dei risultati e supporto alle decisioni
* Gestione delle criticità

1. Problema infeasible
2. Tempo di computazione eccessivo
3. **Strategie e tecniche di ottimizzazione computazionale**

* Warm start
* Parallelizzazione per reparto
* Euristiche e metaeuristiche

1. **Inquadramento e impostazione progettuale**

**Obiettivo.** Il programma genera orari di lavoro ottimali per cliniche e case di cura, rispettando vincoli legali, contrattuali e organizzativi, tenendo conto delle esigenze di copertura dei reparti. I vincoli sono classificati in **HARD** (inderogabili: se non esiste una soluzione che li rispetta il problema viene dichiarato non risolvibile) e **SOFT** (derogabili con relativa penalità in funzione obiettivo).

**Caratteristiche principali:**

* Si può definire l’orizzonte temporale dell’orario da elaborare. I diversi vincoli fanno riferimento a finestre temporali diverse (giorni, settimane, mesi).
* Possibilità di **lock pre-assegnazioni**. Si possono definire assegnazioni dipendente-turno da soddisfare necessariamente oppure da vietare categoricamente.
* Viene garantita la **copertura** richiesta per ogni giorno–reparto–turno.
* Si prevedono **vincoli orari** (massimi mensili/settimanali, numero di giorni consecutivi) e di **riposo** (11h tra turni, riposi settimanali).
* Gestisce i **turni notturni** (idoneità, limiti settimanali/mensili, notti consecutive) e i **pattern post-notte.**
* Considera **assenze individuali** (ferie/malattia/permessi) che vengono conteggiate come ore lavorate per quanto riguarda i vincoli contrattuali.
* Supporta una logica **multi-reparto**: Ogni dipendente è affiliato a un reparto, coperture e orari sono indicati separatamente da ciascun reparto. In caso di necessità e a parità di ruolo, si abilitano assegnazioni in reparti diversi dal proprio, se considerati idonei.
* Bilancia la distribuzione (fairness) di notti, weekend/festivi, **proporzionalmente al FTE** (full-time equivalent) per adeguare proporzionalmente i parametri dei lavoratori part-time.
* Ogni dipendente ha un **saldo ore progressivo** che indica l’accumulo di ore di lavoro in più o in meno rispetto a quelle previste dal contratto. Si cerca, se possibile, di ridurlo.
* **Modalità di pianificazione flessibile:** il modello può essere utilizzato sia per **generare l’intero piano mensile** prima dell’inizio del periodo, sia per **ricalcolare dinamicamente il calendario in corso di mese**, mantenendo il più possibile la stabilità del piano attraverso una penalità che scoraggia modifiche non necessarie rispetto alla versione precedente.

**Scelte di modellazione**

* **DEFINIZIONE DEI TURNI**: I turni sono considerati **contenitori temporali indivisibili**, ne viene sempre preservata l’integrità.  
  Ogni dipendente in una giornata può trovarsi nei seguenti stati:
* **M**: turno del mattino
* **P**: turno pomeridiano
* **N**: turno notturno
* **SN**: smonto notte. Non è un vero e proprio turno, è un indicatore per segnalare che il giorno successivo a un turno notturno non va conteggiato come giornata di riposo. Si attiva solo se il giorno successivo all’inizio del turno notturno non vengono assegnati altri turni. (Quindi se vengono assegnate due notti consecutive la sequenza sarà N-N e non N-SN)
* **R**: giornata di riposo
* **F**: ferie/malattia/permesso/assenza
* Esistono degli orari globali predefiniti per ciascun turno ma ogni reparto può, secondo le esigenze, modificarli. Possono anche essere creati nuovi turni, che possono essere abilitati in tutti i reparti o solo nei reparti che lo richiedono.
* La ***compatibilità ruolo↔turno*** èdefinita globalmente per ruolo. Utile per esempio per abilitare o disabilitare un ruolo ai turni notturni.

Anche per singolo reparto si può disabilitare un turno non richiesto.

Si può inoltre definire la possibilità o impossibilità di coprire un turno notturno anche a livello di singolo dipendente.

* **Copertura per reparto**: Si prevede che ogni reparto richieda per ciascun giorno e per ciascun turno una domanda di personale relativamente a ogni ruolo.
* **Overstaffing:** possibilità di impedire categoricamente l’overstaffing o di permetterlo fissando un tetto.
* La maggior parte dei parametri che vengono utilizzati nell’espressione dei vincoli seguono una logica di **ereditarietà**: valore specifico per dipendente → fallback per ruolo/reparto → fallback globale
* **Bordi dell’orizzonte**: quando viene elaborato l’orario si “guarda indietro” alcuni giorni per applicare correttamente i vincoli che hanno bisogno di controlli sul periodo precedente.
* **Tecnologia di soluzione:** Il modello di ottimizzazione adottato è **OR-TOOLS (CP-SAT)**, ideale per la presenza di variabili di assegnazione binarie e vincoli lineari.

### **Gestione delle assenze successive alla pianificazione:** Nel modello le assenze vengono trattate in modo differenziato a seconda del momento in cui sono comunicate.

### Se l’assenza è **nota prima dell’elaborazione del piano**, la sua durata viene valorizzata in base alle **ore standard previste per il ruolo** del dipendente secondo contratto.

### Se invece l’assenza viene **registrata dopo la generazione dell’orario**, il sistema utilizza le **ore effettive del turno originariamente assegnato** in quella giornata.

1. **Definizione dei Vincoli**

Possono essere definiti i seguenti gruppi principali di vincoli: vincoli di copertura della domanda, vincoli orari, vincoli sul riposo, vincoli sulle notti, vincoli sui pattern, vincoli sul saldo progressivo.

I parametri che fanno riferimento ai vincoli sono stati definiti discrezionalmente e possono essere modificati dall’utente in base alle necessità.

**Vincolo sulla copertura della domanda** di personale per giorno/reparto/turno (**vincolo HARD).**

Per ogni giorno dell’orizzonte di pianificazione, ogni reparto indica nel month plan il suo piano di turni e la relativa domanda di personale, che viene sviluppata sia per gruppi di ruoli che per singolo ruolo.  
Esempio: si può richiedere che siano presenti almeno 7 dipendenti tra infermieri e oss, di cui almeno 2 infermieri e almeno 3 oss.

La domanda di personale deve essere necessariamente soddisfatta.

**Vincoli orari:**

* **Ore mensili teoriche secondo contratto - vincolo SOFT**

Esiste un default per ruolo per le ore previste dal contratto nel mese di pianificazione, con possibilità di override per singolo dipendente.  
Si prevede la possibilità di derogare alle ore previste da contratto se necessario, penalizzando opportunamente la deroga in funzione obiettivo.

* **Ore mensili massime inderogabili - vincolo HARD**  
  Pur permettendo di assegnare più ore di quelle previste da contratto si prevede un tetto che non si può superare. Se non indicato esplicitamente dall’utente, si è scelto di impostarlo di default pari al 25% in più delle ore contrattuali mensili previste.
* **Ore settimanali massime - vincolo HARD**

Si prevede un massimo anche alle ore settimanali per non concentrare tutte le eventuali ore extra mensili in una sola settimana. Si è scelto di impostare il massimo di ore settimanali uguale al 40% in più di quelle teoriche. Può essere modificato in base alle esigenze.

**Vincoli notti**

* **Idoneità notte per ruolo/dipendente** (**HARD)** → (se non idoneo: nessun turno notte assegnabile).
* **Numero massimo di notti per settimana/mese** (**HARD):**

Si definisce per ogni ruolo se siano previsti turni notturni e, in caso positivo, il numero massimo di notti settimanali e mensili. Possibilità di override per singolo dipendente.

* **Numero massimo notti consecutive - vincolo HARD**
* **Penalità per ogni notte consecutiva dopo la prima -> vincolo SOFT**

Vengono permesse notti consecutive (con penalità) ma viene comunque stabilito un massimo che non si può superare.

**Vincoli sui pattern:**

* Dopo una **sequenza di 2 o più notti** deve esserci una **sequenza SN (smonto notte) − R (Riposo)** nei due giorni successivi (o SN-F) - **vincolo HARD**.
* Dopo una singola notte viene suggerita una **preferenza per la sequenza** **SN (smonto notte) − R** - **vincolo SOFT**. (Nel caso in cui il dipendente ha ferie o permessi, non si penalizza nemmeno la sequenza **N-SN-F**)
* **Incompatibilità Notte→Mattina** e **Notte→Pomeriggio** -> **vincolo HARD**
* **Turni notturni incompatibili con assenze** il giorno successivo → **vincolo HARD**  
  Se un giorno D è prevista un’assenza per Ferie/Malattia/Permesso, si vieta l’assegnazione di un turno Notte che termini in D (niente turno N al giorno D−1).

**Vincoli riposo:**

* **Riposo di 11 ore** tra un turno e l’altro **-> VINCOLO SOFT,** valido per tutti i ruoli.
* **Numero massimo di deroghe** alle 11 ore di riposo in un mese **-> vincolo HARD**
* Numero massimo di **deroghe consecutive** alle 11 ore di riposo **-> vincolo HARD**

**Nota**: Anche per questi parametri c’è la possibilità di sovrascrivere il parametro globale e impostarlo per singolo dipendente.

* **Riposo un giorno a settimana -> VINCOLO SOFT** per tutti i ruoli
* **2 giorni di Riposo su 2 settimane -> VINCOLO HARD**   
  Si può derogare al giorno di riposo settimanale, ma in quel caso si deve recuperare il mancato riposo la settimana successiva.

**Nota**: per questi ultimi due vincoli si intende la settimana come finestra mobile. Quindi vengono controllati anche i turni assegnati nei giorni precedenti all’orizzonte di pianificazione per verificare il rispetto dei vincoli.

**Vincoli sul saldo ore progressivo da fine a inizio mese:**

Il modello controlla il bilancio delle ore dovute ed effettivamente svolte per ciascun dipendente su base mensile e cumulata.  
Si prevedono quindi:

* + **vincoli soft**, che penalizzano gli scostamenti dal numero di ore teoriche del mese, in modo proporzionale al saldo iniziale di ciascun dipendente (per chi ha un saldo progressivo positivo si penalizzano ulteriori surplus di ore, e viceversa per chi ha un saldo progressivo negativo).
  + una **penalità globale** sul saldo finale che riduce la dispersione complessiva tra i dipendenti
  + un **vincolo hard** sulla variazione massima mensile, che impedisce correzioni troppo brusche del saldo da un mese all’altro.

**Vincoli di Fairness (notti e weekend/festivi):** Si penalizza la dispersione nel numero di notti e di turni festivi complessivi tra i dipendenti di ciascun reparto (storico + mese corrente), favorendo una distribuzione equa. **(VINCOLO SOFT)**

**Cambi di reparto:**

L’assegnazione di un dipendente a un reparto diverso da quello principale è consentita **solo se il dipendente è considerato idoneo** a operare nel reparto di destinazione.  
L’idoneità viene determinata secondo le seguenti regole:

* **Parità di ruolo:** un dipendente può essere assegnato a turni di un altro reparto solo se appartiene al ruolo professionale richiesto.
* **Pool di reparti abilitati:** per ciascun ruolo viene definito un insieme (***pool***) di reparti nei quali quel ruolo è abilitato a prestare servizio. Le assegnazioni sono consentite esclusivamente all’interno del pool di appartenenza.

L’architettura scelta presuppone quindi che, per ogni ruolo, la compatibilità tra reparti sia simmetrica e transitiva. Se così non fosse, andrebbe sostituita con una matrice di compatibilità per singole coppie (ordinate) di reparti.

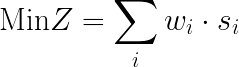
**Stabilità del piano:**

In caso di ricalcolo in corso di mese, il modello penalizza le modifiche rispetto al piano precedente, favorendo la coerenza delle assegnazioni già comunicate e riducendo l’impatto operativo delle revisioni. **(VINCOLO SOFT)**

### **3. FUNZIONE OBIETTIVO E MODELLAZIONE MATEMATICA**

### ***Struttura generale***

La funzione obiettivo del modello è una **somma ponderata di penalità** associate a violazioni o comportamenti indesiderati.  
L’obiettivo è **minimizzare** tale somma, in modo da trovare una pianificazione che rispetti il più possibile le regole e le preferenze organizzative.



dove:

* **si** =  numero di occorrenze di mancato soddisfacimento di una clausola o, se usata una scala temporale, minuti / ore in cui non si rispetta un vincolo o preferenza.
* **wi**= peso (penalità)

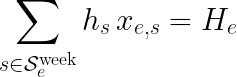
I vincoli hard restano sempre soddisfatti; quelli soft possono essere violati, ma ogni violazione incrementa il valore della funzione obiettivo proporzionalmente al suo peso.

### ***2. Componenti principali della funzione obiettivo***

#### **a. Slack dei vincoli soft**

Per chiarire il significato delle **variabili di slack** è utile mostrare come esse vengono introdotte per trasformare un vincolo rigido in un vincolo “morbido”.  
 Nel caso di un vincolo *hard*, l’uguaglianza o la disuguaglianza devono essere sempre rispettate; quando invece il vincolo viene reso *soft*, si introduce una variabile di slack che misura l’entità della violazione consentita. Tale variabile è poi penalizzata in funzione obiettivo, in modo che il solver tenda comunque a ridurla al minimo.

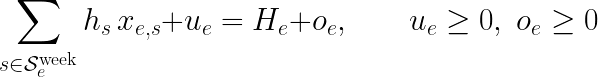
A titolo di esempio, consideriamo il vincolo relativo al **monte ore settimanale per dipendente.** Nel modello *hard*, il totale delle ore assegnate a ciascun dipendente e deve coincidere esattamente con il target settimanale.



Se = insieme di tutti gli slot s in cui il dipendente e può potenzialmente lavorare  
hs = ore di lavoro dello slot s  
xe,s = variabile binaria di assegnazione: vale 1 se il dipendente e viene assegnato allo slot s, 0 altrimenti  
He = ore complessive di lavoro del dipendente e

Nella versione *soft*, invece, vengono introdotte due variabili di slack:

* ue rappresenta le ore in difetto (sotto-copertura) del dipendente **e,**
* oe rappresenta le ore in eccesso (sovra-copertura).



Il vincolo diventa quindi flessibile: può essere violato, ma le violazioni vengono poi **penalizzate nella funzione obiettivo**, ad esempio con un termine di questo tipo:



dove **w-** rappresenta il peso della penalità attribuita alle ore in difetto e **w+** quello alle ora in eccesso. In questo modo si incoraggia il modello a rispettare il più possibile il monte ore previsto per ogni dipendente. Per ora nel programma è stato scelto di impostare i due pesi uguali.

#### **b. Saldo ore progressive**

Per ciascun dipendente il modello gestisce un **saldo progressivo** di ore rispetto al monte ore contrattuale, positivo se ha lavorato più del dovuto, negativo altrimenti. In input viene considerato il saldo progressivo iniziale, accumulato dall’inizio dell’anno fino alla data di inizio dell’orizzonte di pianificazione.  
Durante la pianificazione mensile, il solver cerca di minimizzare gli scostamenti e favorire nel tempo il riequilibrio dei saldi individuali, nel rispetto dei vincoli di copertura, riposi e limiti orari.

### **Penalità mensile con coefficienti adattativi**

Per ogni dipendente con le variabili di slack introdotte (vedi paragrafo precedente) è possibile scrivere le seguenti equazioni di bilanciamento ore:

**Bilancio mensile**



**Aggiornamento saldo progressivo**



#### **Variazione massima mensile (vincolo HARD)**



Questo vincolo limita i recuperi o peggioramenti eccessivi in un singolo mese, garantendo una variazione graduale.

#### **Zona di tolleranza (deadband)**

Per evitare penalità su scostamenti minimi si può introdurre una soglia di tolleranza, in modo che se la differenza tra ore dovute e ore effettivamente svolte risulta essere inferiore a un certo livello non viene penalizzata.



dove d è la soglia di tolleranza espressa in ore.

#### **Coefficienti adattativi e normalizzazione**

I pesi delle penalità mensili dipendono dal **saldo iniziale** individuale e sono normalizzati per garantire un comportamento coerente tra i dipendenti.

* **Clamping degli outlier (10°–90° percentile)**

Si calcolano sui saldi iniziali i percentili Q10 e Q90 e si definiscono i limiti:

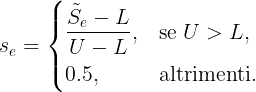


con **Smax***⁡* = cap massimo (es. 60h).

Per il dipendente **e** quindi si ha che il suo saldo progressivo iniziale viene mappato in:



**Normalizzazione lineare (mappa affine)**  
Si ottiene un valore normalizzato se ∈[0,1]:



Si tratta di una **mappa affine**, cioè una retta che riscala l’intervallo [L,U] in [0,1]:

saldi negativi → se≈0; saldi positivi → se≈1

**Coefficiente adattivo**



In questo modo:

* + chi ha **saldo positivo** (in surplus) → penalità **più alta** sull’OVER e **più bassa** sull’UNDER
  + chi ha **saldo negativo** (in deficit) → penalità **più bassa** sull’OVER e **più alta** sull’UNDER

**Normalizzazione per ruolo (opzionale)**  
 Se i ruoli hanno giornate tipo differenti, i coefficienti possono essere divisi per le ore medie giornaliere del ruolo.

#### **Termine mensile in funzione obiettivo**

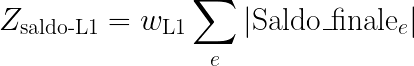
#### 

dove **wover,wunder** sono i pesi base globali per le ore in eccesso e in difetto.

Questo termine è **locale**: penalizza le deviazioni individuali in funzione del saldo iniziale, ma non impone un riequilibrio collettivo.

**Penalità globale sul saldo finale (termine L1)**

Per incoraggiare un **riequilibrio generale dei saldi progressivi**, anche quando tutti i dipendenti potrebbero lavorare esattamente le ore dovute, si introduce un ulteriore termine di penalità basato sulla **somma dei valori assoluti dei saldi** alla fine dell’orizzonte di pianificazione.



che equivale a minimizzare la **norma L1** del vettore dei saldi finali.  
Si sceglie la norma L1 e non la varianza o la norma L2 in quanto così si mantiene la funzione obiettivo **lineare** e facilmente gestibile dal solver CP-SAT.

#### **Significato operativo**

* Spinge tutti i saldiverso **zero**, anche in assenza di vincoli che impongano deviazioni.
* A differenza dei coefficienti adattativi (che agiscono solo quando è necessario generare scostamenti dalle ore dovute), il termine L1 aggiunge una **spinta globale** verso l’equilibrio.

Il beneficio dell’equilibrio deve superare il costo delle ore extra create:



affinché uno scambio di ore da B (saldo positivo) ad A (saldo negativo) sia conveniente.

#### **Scelta del peso wL1**

* Se wL1 è **molto piccolo** (1–3% dei pesi mensili), il termine agisce come *tie-breaker*: riequilibra solo quando è indolore.
* Se wL1  è **più alto** (≈ metà del marginale medio per ora), spinge attivamente il sistema a ridurre i saldi anche quando tutti potrebbero lavorare le ore contrattuali.

#### **Funzione obiettivo completa (saldo progressivo)**

Quindi i termini relativi al saldo progressivo che compaiono in funzione obiettivo sono:



#### **Effetto complessivo**

* Il termine **mensile adattivo** guida le scelte per singolo dipendente, determinando chi deve fare ore in più o in meno.
* Il termine **L1 globale** introduce una preferenza *di sistema* verso la neutralità complessiva dei saldi.
* La combinazione dei due produce un comportamento equilibrato: il solver assegna le ore coerentemente con i saldi individuali e, nel tempo, tende a far convergere l’intero organico verso saldi prossimi allo zero.

**Nota:** Il presente termine della funzione obiettivo è espresso in **ore**, mentre la maggior parte degli altri termini è **adimensionale**, in quanto misura semplicemente il **numero di violazioni** o di occorrenze di una certa clausola.  
Per rendere omogenei i contributi e garantire confrontabilità, le ore vengono **normalizzate** dividendo per la **durata media di un turno** del ruolo di appartenenza.  
 In questo modo il termine non rappresenta più “ore in più o in meno”, ma il **numero equivalente di turni medi in più o in meno**, consentendo di valutare in modo coerente l’impatto delle penalità tra ruoli con diversa struttura oraria.

#### ***c. Penalità per turni in un altro reparto***

Se un dipendente viene assegnato a un reparto diverso da quello di appartenenza (ma eleggibile), si applica una penalità proporzionale al numero di tali assegnazioni:



dove cross(e,s) = 1 se lo slot s appartiene a un reparto diverso da quello principale di e.

***d. Fairness su notti, festivi e weekend***

Si introducono penalità per **squilibri** tra dipendenti su:

* numero di turni notturni
* numero di weekend e festività lavorate

Per ciascun criterio si calcola una misura di dispersione e la si aggiunge come penalità. L’obiettivo è distribuire equamente i turni tipicamente meno desiderati.

Il criterio non impone un vincolo rigido, ma introduce un costo proporzionale alla misura della dispersione.

### **Fairness notti**

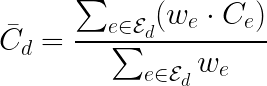
Per ogni reparto, si considerano i soli dipendenti abilitati a coprire turni notturni.  
Per ciascuno, si calcola il numero totale di notti cumulate fino alla fine dell’orizzonte di pianificazione:



dove:

* He  è il numero di notti già effettuate prima dell’orizzonte (storico cumulato);
* Ne è il numero di notti pianificate nel mese corrente.

Si determina quindi la **media pesata di reparto**, utilizzando come pesi i coefficienti **we** (uguale al coefficiente FTE, in modo da scalare opportunamente per i lavoratori part-time)



La deviazione assoluta di ciascun dipendente rispetto alla media è linearizzata tramite variabili ausiliarie ue≥0:

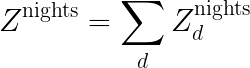


Il termine di penalità per reparto è quindi:



Dove αnights è il peso attribuito alla penalità relativa in funzione obiettivo.

Quindi la penalità complessiva di fairness notti è pari alla somma di quella di ogni reparto:



Tale termine penalizza la **dispersione interna** delle notti totali (storico + pianificate) nel reparto, spingendo i totali individuali a convergere verso la media pesata.  
L’effetto è una redistribuzione progressiva dei turni notturni nel tempo, che tiene conto sia della storia pregressa sia del piano in costruzione.

**Nota**: Nel POC è utilizzato un unico coefficiente globale, αnights , uguale per tutti i reparti. La formulazione è tuttavia generalizzabile introducendo un coefficiente diverso per singolo reparto, utile in caso di pesi differenziati tra reparti con diverso profilo operativo.

### **Fairness weekend/festivi**

Per la distribuzione di turni su **sabato, domenica e festivi**, si adotta una formulazione analoga a quella descritta per le notti.  
In questo caso, per ogni dipendente si considerano i **turni** cumulativi, sommando quelli storici e quelli pianificati.  
La penalità misura la dispersione rispetto alla media pesata dei turni totali di reparto, secondo lo stesso schema di linearizzazione e ponderazione descritto sopra, con coefficiente globale per tutti i reparti.

Formalmente, il termine complessivo di fairness weekend/festivi introdotto in funzione obiettivo è:



dove le variabili uewe rappresentano le deviazioni assolute dei turni cumulativi rispetto alla media pesata del reparto.

### **Effetto complessivo**

I termini *Znights* e *Zwe*vengono sommati alla funzione obiettivo complessiva:



ottenendo una penalità che promuove l’equità nella distribuzione dei turni gravosi (notti e weekend/festivi) all’interno dei singoli reparti.

Il modello rimane interamente lineare e parametrizzabile tramite i coefficienti αnights e γwe, che consentono di regolare il peso relativo del criterio di fairness rispetto agli altri obiettivi.

**Nota:** In prospettiva, potrà essere introdotta una penalità di tipo **min–max** in alternativa o in aggiunta a quella basata sulla dispersione.  
Tale approccio consente di minimizzare direttamente lo sbilanciamento massimo tra i dipendenti in termini di carico di turni notturni o festivi, ottenendo una misura di equità più stringente e intuitiva.

  
dove:

* ***Li*** è il numero (o il peso) complessivo di turni gravosi assegnati al dipendente *i*  
   (ad esempio notti, festivi o weekend);
* ***Max ⁡Li***rappresenta il carico massimo tra tutti i dipendenti;
* ***Min ⁡Li*** rappresenta il carico minimo;
* l’obiettivo è minimizzare lo scarto tra i due estremi, ottenendo una distribuzione più equilibrata dei turni.

***e. Stabilità del piano***

Il modello supporta due modalità di utilizzo distinte:

1. **Pianificazione preventiva**, per la generazione completa del calendario di tutto il mese prima dell’inizio del periodo operativo.
2. **Ricalcolo adattivo**, per aggiornare il piano durante il mese a fronte di variazioni operative (assenze, richieste, disponibilità), mantenendo il più possibile la coerenza con la pianificazione originale.

In quest’ultima modalità, il modello introduce una **penalità di stabilità**, che scoraggia modifiche inutili o eccessive rispetto al piano precedente, privilegiando soluzioni che rispettano il più possibile le assegnazioni già comunicate.

### **Funzione di penalità**

Sia xe,s  la variabile binaria che indica se il dipendente e è assegnato al turno snel nuovo piano, e xe,sold la corrispondente assegnazione nel piano originale.

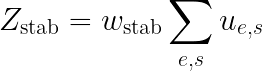
Per ogni coppia, si calcola la differenza assoluta tra i due valori:



Questa variabile vale:

* 0 se il turno rimane invariato;
* 1 se il turno cambia (aggiunto o rimosso rispetto al piano precedente).

La penalità di stabilità è quindi definita come:



dove wstab è un coefficiente di peso che controlla l’importanza del mantenimento del piano rispetto agli altri criteri di ottimizzazione.

### **Effetto del termine**

* In **pianificazione iniziale**, non è presente un piano precedente, quindi il termine non viene applicato. Il modello genera liberamente la soluzione ottima per l’intero mese.
* In **ricalcolo in corso di mese**, il termine viene attivato: il solver viene spinto a modificare solo i turni necessari per migliorare la fattibilità o rispondere a nuovi vincoli, ma viene penalizzato per ogni cambiamento rispetto alla versione precedente.
* È possibile regolare l’entità della penalità tramite il parametro wstab : valori più alti rendono il modello più “conservativo”, mentre valori più bassi favoriscono l’adattamento.

### **Descrizione operativa**

In pratica, l’utente può:

1. **Generare il piano completo del mese**, prima dell’inizio del periodo, ottimizzando copertura, carichi e vincoli di fairness.
2. **Ricalcolare dinamicamente** il piano durante il mese, ad esempio in seguito a:
   1. nuove disponibilità o richieste dei dipendenti;
   2. assenze o sostituzioni;
   3. modifiche del fabbisogno giornaliero.

In questa modalità, il modello tiene conto del piano precedente e introduce la penalità di stabilità, così che i cambiamenti siano limitati e coerenti con la versione già comunicata al personale.

### ***f. Funzione obiettivo complessiva***

Sommando tutti i contributi:



Il modello minimizza **Ztot**  
I pesi vanno calibrati in modo da bilanciare i diversi obiettivi. Tendenzialmente l’indicazione è di assegnare pesi alti con le violazioni che hanno a che fare con l’efficienza operativa e i vincoli contrattuali, bassi per quelli che hanno a che fare con equità e preferenze.

***g. Possibili sviluppi e integrazioni***

Il modello potrà essere progressivamente ampliato per includere ulteriori aspetti della pianificazione e riflettere in modo più accurato la realtà operativa della clinica.  
Tra le principali evoluzioni previste:

* **Richieste di copertura per skill**: introduzione di vincoli che garantiscano, per ciascun turno o reparto, la presenza minima di personale con determinate competenze (ad esempio infermieri specializzati, tecnici di laboratorio, figure di coordinamento). Ciò permetterà di modellare in modo più realistico le esigenze qualitative della copertura.
* **Preferenze dei dipendenti**: integrazione di criteri che tengano conto delle richieste individuali al fine di aumentare la soddisfazione e la sostenibilità organizzativa del piano.
* **Uso di risorse esterne non dipendenti**: possibilità di includere personale a chiamata, collaboratori occasionali o altre risorse temporanee per coprire i turni scoperti o gestire picchi di domanda, con vincoli e penalità dedicati.

Queste integrazioni permetteranno di aumentare il livello di dettaglio e di adattabilità del modello, mantenendo al contempo la sua struttura generale e la compatibilità con le logiche di ottimizzazione già definite.

1. **Valutazione delle soluzioni e gestione delle criticità**

### ***Impostazione dei pesi e valutazione della soluzione***

La definizione dei pesi nella funzione obiettivo deve riflettere la priorità reale dei diversi criteri di pianificazione.  
In generale, ai vincoli di tipo **legale o contrattuale** viene attribuito il peso più elevato, seguiti da quelli di **copertura operativa** e **stabilità del piano**, mentre criteri di **fairness** o **preferenze individuali** assumono pesi inferiori.  
La calibrazione iniziale può basarsi su valori qualitativi (ad esempio, “copertura = molto alta”, “fairness = media”) per poi essere raffinata iterativamente osservando i risultati.

La valutazione di una soluzione non si limita alla verifica della fattibilità, ma considera il **breakdown della funzione obiettivo**, ossia la scomposizione del valore totale nei contributi dei diversi termini (copertura, stabilità, fairness, preferenze, ecc.).  
Questo consente di individuare quali aspetti influenzano maggiormente il costo complessivo e di intervenire in modo mirato sulla pesatura, migliorando progressivamente l’equilibrio tra qualità e tempi di calcolo.

In questo modo, oltre a perseguire l’obiettivo principale di **pianificazione ottimale dell’orario**, il sistema consente di **evidenziare le principali criticità operative** e di supportare **decisioni manageriali consapevoli** in merito all’organizzazione del personale e alla distribuzione delle risorse.

### ***Gestione delle criticità***

Dopo l’esecuzione del processo di ottimizzazione, la soluzione ottenuta viene valutata sia in termini di **qualità operativa** (rispetto dei vincoli e delle preferenze), sia in termini di **efficienza computazionale** (tempi di calcolo e stabilità del risultato).  
 Durante questa fase possono emergere alcune **criticità tipiche**, legate sia alla natura dei dati sia alla complessità del modello:

* il solver può **non trovare alcuna soluzione fattibile** (*problema infeasible*);
* può **richiedere tempi di elaborazione troppo elevati** per convergere verso l’ottimo.

Nei paragrafi seguenti sono descritte le principali cause di tali situazioni e le relative **strategie correttive**, suddivise per tipologia di problema.

1. ***Problema infeasible***

Quando il solver non riesce a trovare alcuna soluzione fattibile, significa che i vincoli **hard** sono tra loro incompatibili o eccessivamente restrittivi rispetto ai dati disponibili.  
 In questi casi è necessario ampliare lo spazio di ricerca attraverso un **rilassamento controllato del modello**.

Le principali azioni correttive sono:

* **Rilassare vincoli secondari o parametri rigidi**, ad esempio estendendo i limiti di riposo, le soglie di ore massime o le regole sui pattern troppo vincolanti.
* **Trasferire vincoli troppo rigidi nei soft**, introducendo penalità elevate ma finite per consentire al solver di individuare comunque una soluzione.
* **Modificare i dati di input** per ampliare il dominio delle soluzioni, ad esempio permettendo a più personale di coprire turni notturni o di operare in più reparti.

### ***2.Tempo di computazione eccessivo***

Quando il solver impiega un tempo elevato per trovare o migliorare una soluzione, il problema non è di incompatibilità tra vincoli ma di **complessità combinatoria** o di **eccessiva ricchezza del modello**.  
 In questi casi è opportuno intervenire per ridurre lo spazio di ricerca o rendere più efficiente la procedura di ottimizzazione, mantenendo comunque una qualità accettabile della soluzione.

Le principali azioni correttive sono:

* **Accettare una soluzione sub-ottimale**, impostando una soglia di accettabilità (ad esempio un *relative gap* del 10%) per interrompere la ricerca quando la qualità è già soddisfacente.
* **Eliminare o semplificare vincoli soft poco significativi**, riducendo la complessità della funzione obiettivo senza compromettere la qualità operativa del piano.
* **Limitare la densità di variabili**, restringendo il numero di dipendenti eleggibili per ciascun turno o reparto, così da contenere il numero di combinazioni esplorate dal solver.

L’obiettivo è garantire tempi di calcolo prevedibili e soluzioni praticabili, anche a costo di sacrificare una piccola parte dell’ottimalità teorica.

1. **Strategie e tecniche di ottimizzazione computazionale**

Anche in presenza di una corretta impostazione dei vincoli e dei parametri, possono emergere **criticità strutturali** legate all’architettura o al flusso del modello di ottimizzazione.  
In questi casi — ad esempio quando il solver impiega costantemente tempi eccessivi o mostra difficoltà di convergenza — è possibile intervenire non solo sui parametri, ma anche **sulla struttura del problema e sul processo di risoluzione**, attraverso modifiche mirate al modello, alla sua decomposizione o alla sequenza di calcolo.  
Tali interventi consentono di migliorare l’efficienza complessiva, garantendo comunque la coerenza con i vincoli e gli obiettivi di pianificazione.

### ***Warm Start***

Un approccio efficace per migliorare i tempi di calcolo e la stabilità delle soluzioni consiste nell’utilizzare una **strategia di warm start**, ovvero fornire al solver un punto di partenza già plausibile rispetto ai vincoli del modello.  
 In questo modo, la ricerca dell’ottimo non parte da uno stato completamente vuoto ma da una configurazione coerente, riducendo il numero di decisioni da esplorare e favorendo la convergenza verso una buona soluzione in tempi più brevi.

Nel contesto del modello di pianificazione per cliniche e case di cura, si considerano principalmente due modalità di warm start, entrambe semplici da implementare e compatibili con la struttura CP-SAT:

1. **Riutilizzo del piano del mese precedente.**  
    Il piano di turni già generato per il mese precedente viene utilizzato come base iniziale per il nuovo orizzonte di pianificazione.  
   Le assegnazioni compatibili (stesso dipendente, stesso turno, stesso reparto) vengono fornite al solver come ***hint*** iniziali.  
   Le giornate o i turni non più validi vengono esclusi, ma la struttura generale del piano rimane simile, consentendo al modello di evolvere da una soluzione “vicina” a quella precedente.  
   Questo approccio risulta particolarmente utile in caso di **ricalcolo in corso di mese**, dove la stabilità e la continuità del piano rappresentano un obiettivo importante oltre alla copertura operativa.
2. **Pattern e sequenze predefinite.**  
   In alternativa, si può inizializzare il modello con **sequenze di turni valide o frequenti**, costruite sulla base di regole operative note (ad esempio, “M–M–R”, “N–SN–R”, “P–P–R”).  
   Tali pattern possono essere applicati a gruppi di dipendenti o a singoli reparti, fornendo una struttura iniziale che rispetta i vincoli di base su riposi e notti consecutive, e riducendo sensibilmente la complessità combinatoria.  
   Il solver completa poi l’assegnazione ottimizzando la copertura e i vincoli soft residui.

Entrambe le modalità sono compatibili con l’architettura attuale del modello e non richiedono modifiche strutturali: il warm start si traduce semplicemente in un set di variabili binarie inizializzate a valori coerenti (solver.SetHint() nel framework OR-Tools).  
Il vantaggio è duplice: **riduzione dei tempi di elaborazione** e **maggiore stabilità delle soluzioni** rispetto ai piani precedenti.

In prospettiva, il warm start potrà essere ulteriormente potenziato fornendo al solver una soluzione iniziale derivata da **euristiche o metaeuristiche** (ad esempio algoritmi greedy, simulated annealing o genetici), che verranno discusse nel paragrafo successivo.

### ***Parallelizzazione per reparto***

Quando il numero di dipendenti e turni cresce, la dimensione del problema combinatorio può aumentare rapidamente, rendendo onerosa la ricerca dell’ottimo anche per un solver efficiente come CP-SAT.  
 Un modo semplice e scalabile per migliorare i tempi di computazione è **suddividere il problema per reparto**, risolvendo in parallelo più istanze indipendenti, e successivamente effettuare una **fase di affinamento globale** che ricompone e ottimizza il piano complessivo.  
 Questa strategia consente di ridurre drasticamente il numero di variabili per singolo modello, sfruttando al contempo la coerenza tra i reparti nella fase finale.

#### **Fase 1 – Ottimizzazione per reparto (in parallelo)**

Per ciascun reparto viene creato un modello autonomo, comprendente solo:

* i dipendenti assegnati a quel reparto,
* i turni e i giorni dell’orizzonte temporale pertinenti,
* i relativi vincoli di copertura, orario e riposo.

Le assegnazioni **cross-reparto sono disabilitate**, così da ridurre il dominio decisionale.  
Tutti i vincoli **HARD** vengono mantenuti, mentre la **copertura della domanda** è trattata come **vincolo SOFT con penalità elevata**, consentendo al solver di trovare una soluzione anche in caso di carenze locali di personale.  
Gli altri vincoli SOFT (fairness, ore progressive, riposi, ecc.) mantengono le stesse pesature previste nel modello globale, garantendo coerenza tra le due fasi.

Ogni istanza per reparto viene risolta **in parallelo**, con un **time limit ridotto** a seconda delle dimensioni e un **relative gap** di accettabilità, ad esempio del 5–10%.  
L’obiettivo non è ottenere la soluzione perfetta per ogni reparto, ma una configurazione coerente e quasi completa che rispetti la maggior parte dei vincoli locali.

Al termine della fase, per ciascun reparto vengono esportate le assegnazioni trovate (x[e,s] = 1) e utilizzate come **warm start** (hint) per la successiva ottimizzazione globale.  
In caso di reparti con soluzione parziale o infeasible, vengono comunque mantenute le assegnazioni valide: il solver della fase successiva potrà completare o correggere la pianificazione grazie alle assegnazioni cross-reparto.

#### **Fase 2 – Affinamento globale**

Nella seconda fase viene lanciato un modello complessivo che include tutti i reparti e abilita le **assegnazioni cross** secondo le regole di idoneità definite nei pool di compatibilità.  
 La **copertura della domanda** torna ad essere un **vincolo HARD**, e vengono ricalcolati tutti i termini globali della funzione obiettivo (saldo ore progressivo, fairness, penalità cross e stabilità del piano).  
 Le soluzioni ottenute nella fase precedente vengono fornite al solver come *hint iniziali* tramite la funzione SetHint() di OR-Tools, migliorando sensibilmente la convergenza.

In questa fase la **stabilità** viene mantenuta attiva con un peso intermedio, così da favorire la coerenza con le assegnazioni già trovate ma senza impedire le correzioni necessarie.  
 Il solver viene eseguito con un **time limit più ampio** (60–180 secondi) e un **gap obiettivo** ridotto (3–5%), permettendo di affinare la soluzione complessiva e garantire una copertura completa e bilanciata tra i reparti.

#### **Strategia alternativa – Doppia run sequenziale (cross disabilitato → abilitato)**

Un’alternativa più semplice dal punto di vista implementativo consiste nell’eseguire due **run sequenziali** dello stesso modello, senza creare istanze separate per reparto.  
Nella **prima esecuzione**, le **assegnazioni cross-reparto vengono disabilitate**, limitando il dominio decisionale ai soli dipendenti e turni del proprio reparto.  
In questa fase la **copertura è vincolo HARD**, ma il solver opera su un problema molto più piccolo e strutturato, ottenendo rapidamente una soluzione “chiusa” per reparto.

Nella **seconda esecuzione**, le variabili cross vengono riattivate e la soluzione precedente viene fornita come **warm start**.  
Il solver può così ottimizzare globalmente, consentendo spostamenti controllati tra reparti per migliorare la copertura complessiva, l’equilibrio dei carichi e il rispetto dei vincoli soft.  
Questa strategia è meno efficiente in termini di tempo complessivo rispetto alla parallelizzazione, ma risulta più facile da integrare poiché utilizza un unico modello e non richiede la gestione di più istanze in parallelo.

#### **Affinamento locale su zone di sotto-copertura**

Dopo l’esecuzione della fase iniziale di pianificazione (sia essa per reparto o globale), può rimanere qualche giorno scoperto o con copertura parziale.  
Per correggere in modo mirato queste situazioni senza ricalcolare l’intero piano, è possibile applicare una **fase di affinamento locale**, in cui il solver opera solo su un sottoinsieme ristretto di variabili.

In pratica, si individuano i **giorni e reparti con sotto-copertura** e, per ciascuno, si **sbloccano** le variabili di assegnazione relative solo ai **dipendenti dello stesso ruolo** in un piccolo intervallo di giorni contigui.   
Tutte le altre variabili restano fissate ai valori trovati nella soluzione precedente, mantenendo così la stabilità del piano.

Il solver può così riassegnare i turni solo all’interno di queste finestre locali, chiudendo i buchi residui e riequilibrando i carichi in modo coerente con i vincoli di riposo, orario e copertura.  
Questa procedura riduce drasticamente lo spazio di ricerca e consente di ottenere un piano pienamente coperto con **tempi computazionali molto contenuti**, evitando modifiche superflue alle parti già valide della pianificazione.

***Euristiche e metaeuristiche***

La pianificazione dei turni può essere affrontata anche servendosi di **tecniche euristiche e metaeuristiche**, utilizzate per ridurre i tempi di calcolo o migliorare la qualità delle soluzioni in problemi complessi e di grande scala.  
Questi metodi non garantiscono l’ottimo matematico, ma ricercano in modo intelligente e controllato soluzioni di buona qualità, bilanciando precisione, rapidità e flessibilità.

Le principali tecniche applicabili al contesto di scheduling sanitario sono le seguenti:

* **Greedy + Repair**  
  Costruisce rapidamente una soluzione iniziale seguendo una logica di priorità — ad esempio assegnando prima i turni più critici o le risorse più vincolate — per poi correggere progressivamente eventuali violazioni o squilibri.  
  È un approccio molto efficace per ottenere **soluzioni fattibili in tempi brevissimi**, che possono essere successivamente migliorate dal solver principale.  
   Viene spesso utilizzata come fase preliminare (*warm start*) per fornire al modello un punto di partenza realistico e coerente con i vincoli operativi.
* **Large Neighborhood Search (LNS)**  
  Parte da una soluzione valida e ne migliora iterativamente la qualità, “rilassando” porzioni limitate del problema (ad esempio una settimana, un reparto o un gruppo di ruoli) e ottimizzandole localmente.  
  Il principio è simile a quello di un affinamento progressivo: anziché risolvere nuovamente l’intero modello, si concentra l’ottimizzazione solo su sottoinsiemi significativi, mantenendo fissi gli altri elementi.  
  Questa strategia, ampiamente utilizzata in combinazione con CP-SAT, consente di **ridurre drasticamente lo spazio di ricerca** e ottenere miglioramenti consistenti in tempi contenuti.
* **Simulated Annealing (SA)**  
  Si ispira al processo fisico di ricottura dei metalli, in cui un materiale viene scaldato e poi raffreddato lentamente per raggiungere una configurazione stabile.  
  In modo analogo, l’algoritmo introduce modifiche casuali alla soluzione corrente e accetta temporaneamente anche peggioramenti, con una probabilità che diminuisce nel tempo.  
  Questo meccanismo consente di evitare di restare intrappolati in minimi locali, migliorando la **capacità esplorativa** del processo e favorendo l’equilibrio tra efficienza e stabilità del piano.
* **Tabu Search (TS)**  
   È una strategia di miglioramento locale che utilizza una *memoria delle mosse* già esplorate (la cosiddetta *tabu list*) per evitare di ripetere schemi di assegnazione recenti e favorire l’esplorazione di nuove configurazioni.  
   L’approccio è più strutturato e deterministico rispetto al Simulated Annealing, e consente una ricerca mirata del miglioramento.  
   È particolarmente efficace come **fase di rifinitura** per ottimizzare criteri di fairness, stabilità o bilanciamento, senza compromettere la fattibilità dei vincoli hard.
* **Genetic Algorithm (GA)**  
   Si basa sui principi dell’evoluzione biologica: una popolazione di soluzioni viene fatta “evolvere” attraverso operazioni di selezione, crossover e mutazione.  
  Le soluzioni migliori vengono combinate per generare nuove configurazioni, con l’obiettivo di ottenere piani progressivamente più equilibrati e robusti.  
  È una tecnica potente per **esplorare ampie regioni dello spazio delle soluzioni** e affrontare obiettivi multipli, come la copertura, la fairness e la stabilità, anche se richiede tempi di calcolo più elevati e un’implementazione più articolata.