

POLITECNICO DI TORINO

Corso di Laurea Magistrale

in Ingegneria Gestionale

Tesi di Laurea Magistrale

Schedulazione degli slot operatori elettivi soggetta ai vincoli di capacità della terapia intensiva post-operatoria



Relatore
Prof. Federico Della Croce di Dojola

Candidata
Stefania Pinna

Anno Accademico 2023/2024

*Ai miei nonni
Stefano e Giacomina
Nannino e Teresa, esempi di vita.*

*Ai miei genitori che mi sostengono
con il loro affetto incondizionato.*

INDICE

Introduzione.....	5
1. Il Servizio Sanitario Nazionale.....	7
1.1. Nascita e sviluppo del Servizio Sanitario Nazionale	7
1.2. Canali di accesso all'assistenza sanitaria.....	12
2. Il Blocco Operatorio.....	14
2.1. Le strutture dell'Azienda Ospedaliero Universitaria di Sassari.....	14
2.2. Le liste d'attesa	18
2.3. I posti di terapia intensiva post-operatoria.....	20
2.4. Le risorse infermieristiche	22
3. La schedulazione delle sale operatorie.....	24
3.1. Revisione della letteratura.....	24
3.2. Osservazioni e criticità dell'attuale gestione	25
3.3. Un approccio costruttivo scritto in codice C++	27
3.4. Un approccio di Programmazione Lineare e Ottimizzazione Combinatoria.....	38
3.5. Descrizione del modello di programmazione lineare del problema proposto	39
3.6. Descrizione del Modello Mosel	42
4. Risultati.....	46
4.1. Analisi e confronto dei risultati ottenuti	46
4.1.1. Output grafici ottenuti dall'algorithmo costruttivo in C++.....	47
4.1.2. Output grafici generati da Mosel.....	50
4.1.3. Istanze generate	51
4.2. Limiti delle soluzioni proposte	54
Conclusioni.....	55

Appendice.....	56
Bibliografia/Sitografia	94

Introduzione

L'offerta dei servizi sanitari italiani sta riscontrando una molteplicità di problematiche legate alla difficoltà di accesso alle cure a causa di liste di attesa infinite che portano milioni di cittadini a rivolgersi a cure a pagamento e, talvolta, alla migrazione sanitaria.

La causa di queste limitazioni nelle cure nella Sanità pubblica è legata principalmente alla carenza di personale sanitario e infermieristico.

L'articolo 32 della Costituzione Italiana recita che “La Repubblica tutela la salute come fondamentale diritto dell'individuo e interesse della collettività, e garantisce cure gratuite agli indigenti”, attribuendo la responsabilità della tutela della salute allo Stato e alle Regioni [1].

I tre principi cardine del Sistema Sanitario Nazionale (SSN) italiano sono l'universalità, l'equità e la solidarietà, al fine di garantire a tutta la popolazione l'accesso alle prestazioni sanitarie a parità di bisogni [2].

Purtroppo, ciò spesso non avviene.

Per far luce sulle cause, è importante considerare il periodo storico nel quale stiamo vivendo: le liste d'attesa registrano un ritardo cronico, aggravato dall'impatto che la pandemia ha avuto negli ospedali e ulteriormente amplificato dalle dinamiche demografiche (popolazione sempre più vecchia, calo delle nascite crescente), dalle strutture talvolta poco adeguate, da una scarsa cultura dell'informatizzazione del dato e dal problema dell'assenteismo registrato tra il personale sanitario [3].

L'obiettivo di questo elaborato di tesi è quello di massimizzare l'utilizzo dei posti in terapia intensiva post-operatoria (TIPO), che richiedono strutture specifiche e la supervisione del personale infermieristico con rapporto infermiere-paziente di 1:1 e che, per questo motivo, dispongono di posti limitatissimi che necessitano di essere ottimizzati nel loro utilizzo.

Una volta programmati gli interventi dei pazienti, l'obiettivo è quello di riuscire a fornire una pianificazione che tenda ad avvicinarsi all'utilizzo ottimale delle sale operatorie, facendo ruotare le specialità chirurgiche senza sforare la capacità consentita.

Le soluzioni proposte prevedono una schedulazione, potenzialmente a medio termine, al fine di adattare la gestione delle risorse, umane e non, al piano di schedulazione, e non viceversa.

Dopo aver introdotto in linea generale il contesto nel quale si sviluppa lo studio e lo scopo del lavoro stesso, la trattazione della tesi si articola su quattro capitoli.

- Il primo approfondisce il contesto ospedaliero e sanitario e fa chiarezza su alcune questioni di cui, spesso, si parla senza reale conoscenza dei fatti.
- Il secondo capitolo focalizza l'attenzione sull'Ospedale di Sassari e le strutture coinvolte, presentando le criticità ed eventuali proposte di risoluzione, relativamente alla gestione delle liste d'attesa, della terapia intensiva post-operatoria e delle risorse infermieristiche.
- Il terzo capitolo presenta le criticità dell'attuale gestione e le proposte di risoluzione, affrontando la descrizione degli algoritmi realizzati per risolvere il problema.
- Il quarto e ultimo capitolo analizza e confronta i risultati ottenuti e i limiti delle soluzioni proposte.

1. Il Servizio Sanitario Nazionale

1.1. Nascita e sviluppo del Servizio Sanitario Nazionale

Il Servizio Sanitario Nazionale sostituì il sistema mutualistico con la legge del 23 dicembre 1978 n. 833, su proposta del ministro della sanità Tina Anselmi, sotto il governo Andreotti IV [4].

La Sanità, dunque, iniziò ad essere considerata come un bene universalmente fruibile dalla Nazione, grazie al Fondo Sanitario Nazionale che incaricò il Governo centrale di fornire le risorse.

Negli anni Ottanta il concetto di salute passò da bene universale e gratuito a bene necessario per l'equità con lo scopo di permettere l'accesso alle cure delle persone meno abbienti.

Nell'articolo 32 della Costituzione si ha: "La Repubblica tutela la salute come fondamentale diritto dell'individuo e interesse della collettività, e garantisce cure gratuite agli indigenti".

L'articolo 117, comma 2, lettera m, da canto suo, prevede che la determinazione dei livelli essenziali delle prestazioni concernenti i diritti civili e sociali debbano essere garantiti su tutto il territorio nazionale [5].

La responsabilità della tutela della salute è ripartita tra lo Stato e le Regioni.

La funzione di ogni articolazione del Servizio Sanitario è quella di garantire ai cittadini, attraverso il riparto alle Regioni delle risorse del Fondo Sanitario Nazionale (FSN), l'accesso uniforme ai Livelli Essenziali di Assistenza (LEA), sulla base di requisiti di qualità, efficienza, appropriatezza, trasparenza e informazione [2].

L'efficienza, il buon funzionamento e la capacità di rispondere alla richiesta del fabbisogno di salute diventa, quindi, il parametro sul quale valutare ogni articolazione del Servizio Sanitario Nazionale.

Negli anni Novanta le strutture pubbliche si trasformarono da Unità Sanitarie Locali (USL) in Aziende Sanitarie Locali (ASL) sulla base della legge delega 23 ottobre 1992, n. 421 [6].

Queste ultime sono organizzazioni sanitarie a livello territoriale che si occupano di fornire assistenza sanitaria primaria, prevenzione, diagnosi e trattamenti ambulatoriali nella regione di appartenenza. Le ASL, che hanno sostituito le USL, sono divise in distretti sanitari.

Le ASL sono affiancate alle Aziende Ospedaliere (AO) che svolgono un ruolo complementare, nello specifico sono strutture ospedaliere ad alta specializzazione e si occupano della gestione di ospedali specializzati e complessi che offrono servizi di alta qualità e trattamenti avanzati. [2]

La differenza sostanziale tra Azienda Ospedaliera e azienda ASL consiste nel fatto che, mentre nella prima l'Ospedale coincide con l'azienda stessa, nella seconda l'Ospedale è solo una delle tante strutture di cui si compone la ASL, insieme agli ambulatori e ai Distretti.

L'Azienda Ospedaliera ha un Direttore Generale, nominato dalla Regione tra gli iscritti tenuti in un apposito elenco presso il Ministero della salute [2].

Il Direttore Generale, che ha l'incarico di gestire l'azienda, nomina, a sua volta, il Direttore Amministrativo e Sanitario [6].

La ASL ha, invece, un Direttore Sanitario e uno Amministrativo che rispondono al Direttore Generale della ASL stessa [6].

La logica dietro questo drastico cambiamento consiste nel far gestire la sanità da aziende private al fine di prestare attenzione al costo, al risultato e alla qualità del servizio offerto e, dunque, ridurre gli sprechi.

Poiché le strutture utilizzate sono pubbliche, ci si è posti il problema dell'utilizzo delle stesse da parte di medici dipendenti della ASL per lo svolgimento di attività private.

Il decreto legislativo 19 giugno 1999 n. 229 ha imposto il divieto di svolgere attività privata all'interno delle strutture pubbliche, attività comunemente chiamate di intramoenia, e l'obbligo di scelta fra una delle due tipologie di attività [3][6].

Questo decreto è stato abolito dopo tredici anni dal decreto Balduzzi, che ha, inoltre, introdotto una differente riorganizzazione del Servizio Sanitario in Italia e la regolamentazione dell'attività medica e scientifica [7].

Ad oggi il problema delle operazioni intramoenia è un argomento che fa discutere.

Le Attività Libero Professionali Intramoenia (ALPI) sono prestazioni offerte in regime ambulatoriale e interventi chirurgici in regime di ricovero ordinario, Day Hospital o Day Surgery [8].

Tali prestazioni nascono con lo scopo di permettere al paziente di scegliere il medico dal quale essere visitato e/o operato, previo pagamento di un costo stabilito a livello nazionale, a carico dell'utente.

Sono prestazioni svolte dal personale della dirigenza medica e sanitaria in regime di esclusività.

La gestione di queste prenotazioni deve essere separata dalle prestazioni istituzionali erogate a carico del SSN, come previsto dalle Linee Guida Nazionali del CUP [8][9].

Le Regioni hanno l'obbligo di controllare i volumi di tali prestazioni al fine di garantire il corretto rapporto tra le prestazioni erogate in libera professione e quelle istituzionali [6].

I Livelli Essenziali di Assistenza sono prestazioni e servizi che il Servizio Sanitario Nazionale è tenuto a fornire a tutti i cittadini, gratuitamente o previo pagamento di ticket [6][17].

Questo servizio è garantito mediante l'utilizzo di una quota parte della fiscalità generale (imposte). L'assistenza istituzionale garantisce l'assistenza dei pazienti entro tempi prestabiliti, sulla base della priorità del caso [2]:

- priorità A: entro 30 giorni dall'immissione in lista d'attesa;
- priorità B: entro 60 giorni dall'immissione in lista d'attesa;
- priorità C: entro 90 giorni dall'immissione in lista d'attesa;
- priorità D: entro 180 giorni dall'immissione in lista d'attesa.

Qualora vengano superati i tempi previsti dal Piano Nazionale di Governo Liste di attesa, l'Azienda è sanzionabile per inadempienza: le prime attività che vengono sospese, a data da destinarsi, sono le operazioni in ALPI.

Nel contesto oggetto di questa tesi, ad esempio, per usufruire delle prenotazioni in regime di intramoenia e poter scegliere una visita col medico scelto, ad esempio un chirurgo, il paziente dovrà chiamare il numero verde presente sul sito dell'AOU di Sassari.

Nella giornata stabilita, il chirurgo visiterà il paziente, stabilendo se necessiti di un intervento.

In caso affermativo il chirurgo lo inserirà nella lista d'attesa (comune con le prenotazioni avvenute tramite CUP) e assegnerà un codice priorità all'intervento.

Poiché il regime di ALPI non deve essere inteso come uno strumento per “scalare” le liste d’attesa, bisogna far riferimento alla legge 3 agosto 2007, n. 120, che dà disposizioni in materia di attività libero-professionale intramuraria e altre norme in materia sanitaria [9][10].

Il paziente in regime di ALPI può essere operato, sulla base dei tempi stabiliti per priorità, come segue:

- paziente di priorità A: può essere operato a partire dal 31esimo giorno dall’immissione in lista operatoria;
- paziente di priorità B: può essere operato a partire dal 61esimo giorno dall’immissione in lista operatoria;
- paziente di priorità C: può essere operato a partire dal 91esimo giorno dall’immissione in lista operatoria;
- paziente di priorità D: può essere operato a partire dal 181esimo giorno dall’immissione in lista operatoria.

Per evitare situazioni spiacevoli e scorrette, la Direttrice della S.S.D. Operation Manager & Operating Room Manager, responsabile del Blocco Operatorio, supervisionerà ogni richiesta di intervento in ALPI.

Ogni chirurgo, previo accordo con il paziente, manderà una mail con l’avanzamento della richiesta di occupazione della sala in una determinata data e slot orario.

Dopo essersi accertata della regolarità della richiesta, sulla base delle disponibilità del giorno, la proposta verrà accettata o rigettata, in quest’ultimo caso con adeguata motivazione, qualora non sia stato rispettato l’iter richiesto o manchi la documentazione prevista.

Prima della legge del 2007 e prima della supervisione dalla parte della Direttrice, invece, il chirurgo, dopo aver visitato il paziente e dopo averne ritenuto necessaria l’operazione, si attivava per operare il paziente nel primo slot disponibile assegnato alla propria specialità, indipendentemente dalla priorità.

Affinché questa attività risulti effettuata in maniera lecita è, dunque, necessario allineare i tempi di attesa di visite e/o operazioni svolte in intramoenia e dei tempi istituzionali.

Ciò significa che, a parità di visita/intervento, un paziente già immesso nella lista di attesa debba essere visitato prima del paziente registratosi successivamente ad un servizio di intramoenia.

Affinché questo servizio avvenga nella maniera corretta, le attività intra moenia vanno effettuate all'esterno della schedulazione giornaliera e vanno richieste al responsabile che approverà o rifiuterà la proposta dopo adeguata motivazione.

È chiaro come l'attività intramoenia non abbia lo scopo di saltare la lista d'attesa ma semplicemente di consentire al paziente la scelta del chirurgo.

Una parte del ricavato andrà all'ospedale che ospita tale attività mentre la parte restante andrà a sostenere i compensi dell'equipe che si è occupata di quell'operazione.

Il problema principale è legato al fatto che spesso, invece, l'attività intramoenia viene utilizzata in maniera irregolare: i pazienti sono portati, talvolta dagli stessi chirurghi, a prenotare operazioni proprio in questo regime, in quanto viene ritenuta l'unica strada possibile per risolvere i propri problemi in tempi ridotti, evitando di passare tramite CUP, il che richiederebbe tempi dilatati, a causa dell'inserimento dell'operazione stessa in lista d'attesa.

Con queste premesse, l'oggetto di questa tesi mira alla riduzione delle liste di attesa, sia nell'incentivo di velocizzare le stesse liste, grazie allo svolgimento di operazioni che avrebbero tardato ad essere svolte, sia in un'ottica di copertura economica dello stesso ospedale il quale, grazie a un'operazione, può recuperare parte del denaro speso per altre.

Talvolta si sente parlare anche di prestazioni aggiuntive che nulla hanno a che vedere con l'attività privata.

La differenza sostanziale tra intramoenia e prestazioni aggiuntive risiede nel fatto che le ultime hanno come unico scopo quello di smaltire più velocemente la lista d'attesa.

Le prestazioni aggiuntive non consentono al paziente la scelta del chirurgo e sono gratuite per il paziente: il compenso destinato all'equipe è stabilito a livello Nazionale ed è distribuito dall'azienda.

Ogni ospedale ha un tetto di spesa massimo destinato a queste prestazioni, stabilito dallo Stato.

Il compenso previsto per le prestazioni aggiuntive, nettamente inferiore a quello ricevuto dalle prestazioni in ALPI, ammonta a 80 €/ora per la dirigenza e a 50 €/ora per il comparto [8][11].

Questo modus operandi può permettere, inoltre, di ridurre il numero di slot operatori liberi, in quanto le prestazioni aggiuntive vengono programmate come extra rispetto al programma operatorio settimanale.

1.2. Canali di accesso all'assistenza sanitaria

I canali di accesso all'assistenza sanitaria italiana sono prevalentemente quattro: Medico di Medicina Generale (MMG), Guardia Medica (GM), Pronto Soccorso (PS) e Medico Specialista.

I primi due accessi sono considerati a bassa complessità, gli accessi dei pazienti a visite specialistiche e/o al pronto soccorso, invece, sono considerati ad alta complessità.

Un tipico accesso all'assistenza sanitaria comprende sia canali di competenza territoriale che ospedaliera: il paziente può accedere a cure sanitarie territoriali se si rivolge alla GM, al MMG se soffre di una patologia cronica; accede, invece, a cure di competenza ospedaliera qualora si rivolga al PS o venga ricoverato in regime ordinario, di Day Hospital o di Day Service [2][12].

La cura del paziente avviene seguendo i seguenti passaggi:

1. Contatto iniziale: il paziente che necessita di cure mediche non urgenti o di consulenza generale è tenuto a rivolgersi al Medico di Medicina Generale [13]. È necessario che segua le disposizioni del medico scelto: generalmente si è soliti contattare il medico tramite chiamata o mail.
2. Esame o consulenza: il paziente viene visitato e vengono valutate le sue condizioni. Il MMG può decidere di prescrivere una terapia o di richiedere un ulteriore parere di un medico specialista. Generalmente questa fase è da considerarsi critica poiché il Medico di base a volte può richiedere più consulenze del dovuto, creando un collo di bottiglia nello step seguente.
3. Esame specialistico: qualora venga richiesta una visita specialistica dal MMG, il paziente viene visitato in ospedale o in un ambulatorio specializzato nella diagnosi e nel trattamento necessario per la cura del paziente. Successivamente lo specialista può ritenere necessario chiedere ulteriori visite o esami specialistici o ritenere

sufficienti gli esami svolti per la valutazione del paziente. Il servizio che consente di prenotare visite mediche specialistiche, esami diagnostici e interventi chirurgici differibili è il CUP (Centro Unico di Prenotazione). La visita specialistica viene erogata in seguito alla prenotazione telefonica e/o tramite il sito web dell'ente sanitario locale. In queste sedi vengono concordate data e ora, successivamente viene inviato un SMS di riepilogo della prenotazione. Il paziente può accedere ai servizi ospedalieri senza il coinvolgimento del medico di famiglia se e solo se la sua condizione è da definirsi critica.

4. Accesso ospedaliero: il paziente che necessita di cure immediate o di natura più gravi viene indirizzato dal MMG al PS, dove verrà trattato in base alla sua condizione. Qualora fosse necessario, il paziente può essere indirizzato in un determinato reparto per ricevere cure specialistiche o interventi urgenti. Il ricovero, infatti, consente di accedere alla diagnostica di secondo e terzo livello senza alcuna impegnativa, bypassando il CUP qualora la sua condizione venga considerata urgente.
5. Follow-up: dopo il trattamento ospedaliero, il paziente viene ripreso in cura dal MMG che ha l'obbligo di monitorarne la condizione di salute.

2. Il Blocco Operatorio

2.1. Le strutture dell'Azienda Ospedaliero Universitaria di Sassari

Le strutture ospedaliere in capo all'azienda ospedaliero universitaria di Sassari (AOU SS) sono quattro [14]:

1. Le Cliniche San Pietro, Sassari;
2. Il blocco del Materno-infantile, Sassari;
3. L'Ospedale Civile Santissima Annunziata, Sassari;
4. L'Ospedale Marino, Alghero.

Dette strutture sono guidate dalle decisioni dei medesimi organi aziendali e strategici, seppur non usufruiscano di risorse condivise; in particolare, l'AOU di Sassari è organizzata secondo uno schema piramidale, seguendo il livello di responsabilità [14][16].

- Il Direttore Sanitario DS è responsabile del Dipartimento Assistenza Integrata D.A.I. e del Dipartimento Professioni Sanitarie.
- Il Direttore Amministrativo DA è a capo del Dipartimento Amministrativo e Tecnico.
- Il Direttore Generale è a capo degli Organi Aziendali, degli Organi Collegiali, dell'Area Staff della Direzione Strategica e collabora con il DA e DS.

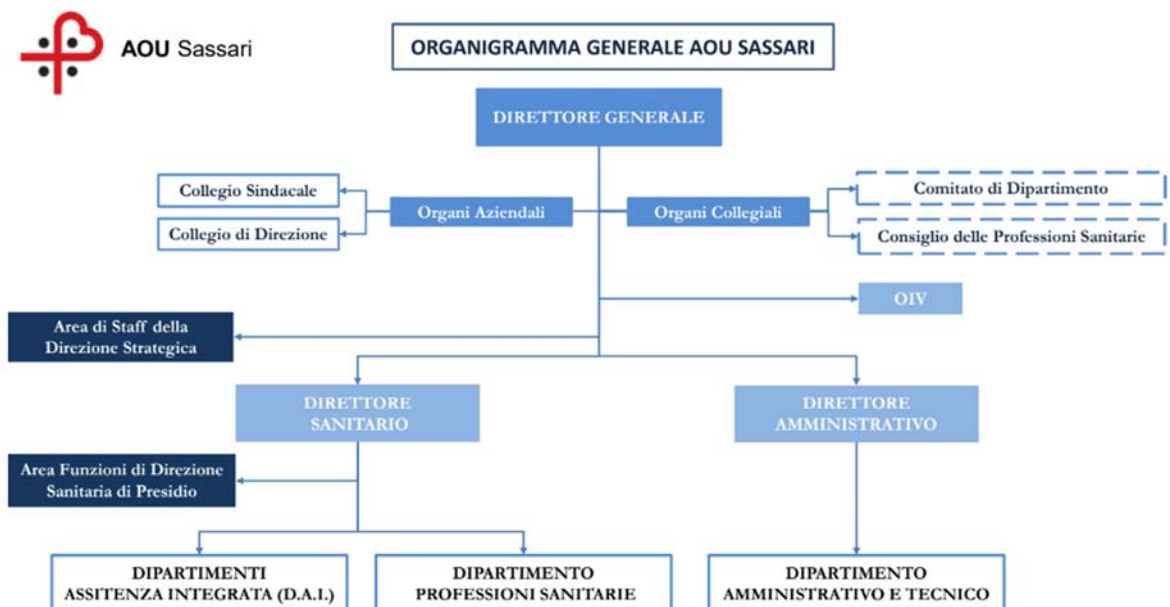


Figura 1: Organigramma generale AOU di Sassari

L'elaborato di tesi si focalizza in particolare sull'analisi relativa alla prima struttura citata, conosciuta anche come "Stecca Bianca" che ospita il blocco operatorio la cui funzione è quella di consentire la regolare esecuzione di interventi chirurgici in regime di elezione e d'urgenza, garantendo le condizioni di sicurezza e sterilità al paziente ed al personale sanitario.

L'attività chirurgica del blocco operatorio si svolge attualmente in 8 sale operatorie nelle quali vengono svolte le seguenti attività specialistiche [14]:

1. Breast Unit;
2. Chirurgia Generale;
3. Chirurgia Maxillo Facciale;
4. Chirurgia Odontoiatrica;
5. Chirurgia Oculistica;
6. Chirurgia Otorinolaringoiatrica;
7. Chirurgia Ortopedica;
8. Chirurgia Pediatrica;
9. Chirurgia Plastica;
10. Chirurgia Urologica;
11. Chirurgia Vascolare;
12. EBUS (Endo Bronchial Ultra Sound)
13. Endoscopia Digestiva;
14. Piede Diabetico;
15. TARF (Termoablazione percutanea dei tumori del fegato);
16. Terapia Antalgica.

Le sale operatorie, numerate dalla 1 alla 8, sono dotate di una presala con relativo locale adiacente, usato per il materiale.

In ogni sala sono presenti due finestre a ghigliottina, saliscendi, una per l'uscita del materiale sporco, l'altra per l'introduzione del materiale sterile.

Questo ospedale si occupa prevalentemente delle operazioni in regime di elezione e talvolta occupa le sale con pazienti che arrivano in emergenza e/o urgenza.

Una delle sale è dotata di un robot che consente di svolgere operazioni ad alta complessità seguendo metodologie poco invasive.

Il blocco operatorio è composto da un insieme di locali articolati in zone progressivamente più sterili, partendo dall'ingresso fino alle sale operatorie.

Al fine di mantenere la sterilità all'interno del comparto operatorio vi sono dei percorsi differenziati tra ingresso, uscita e zona filtro.

I locali su cui si divide il blocco operatorio sono:

- spogliatoi, ove il personale si cambia per accedere al comparto, e servizi igienici per il personale;
- zona filtro all'ingresso;
- magazzino;
- locale sterilizzazione materiale e deposito materiale sterile;
- stanza per la preparazione dei farmaci di anestesia;
- otto sale operatorie;
- una Recovery Room.

Il blocco operatorio per mantenere la sterilità ambientale è isolato verso l'esterno da una serie di zone filtro poste lungo il percorso di avvicinamento alle sale operatorie.

Gli ospedali sono luoghi in cui la pulizia e l'igiene sono indispensabili per prevenire la diffusione di infezioni e garantire la sicurezza dei pazienti e del personale, è pertanto necessario definire i percorsi sporco-pulito e quindi strategie e procedure attuate per gestire movimenti del personale, dei materiali e dei rifiuti all'interno delle strutture sanitarie.

Le aree che collegano gli ingressi alle sale operatorie fanno parte dei cosiddetti "percorsi sporchi" o "contaminati" e sono via via più sterili nel momento in cui si accede alla zona di filtro delle singole sale operatorie e sono totalmente sterili all'interno della sala.

Al fine di garantire la sicurezza dei pazienti e allo scopo di isolare persone potenzialmente infette o contaminate, ogni sala operatoria è progettata per funzionare in sicurezza grazie all'utilizzo di misuratori di pressione collegati a un sistema di pressione negativa.

La "pressione negativa" si ottiene grazie ad un'apparecchiatura che richiama l'aria dall'interno verso l'esterno a basso e medio flusso [41].

L'aria interna alla sala fuoriesce mediante un filtro.

Questo sistema ha permesso una adeguata gestione dei pazienti positivi al Covid durante la Pandemia e consente tutt'oggi una gestione ottimale di pazienti affetti da infezioni tradizionali e microbatteri.

Ogni blocco operatorio dell'AOU di Sassari è provvisto di una propria centrale di sterilizzazione, garantendo piena autonomia alla struttura.

La sterilizzazione del kit chirurgico avviene secondo tre modalità che hanno tempi ciclo differenti e garantiscono una durata dello strumento sterile variabile:

- sterilizzazione a vapore: tempo di sterilizzazione pari a 45 minuti, kit sterile fino a 30 giorni;
- sterilizzazione a vapore flash: è utilizzata per sterilizzare strumenti ad alta rotazione, generalmente per lo strumentario dell'Oculistica Clinica e ha un tempo di decontaminazione 12 minuti;
- sterilizzazione chimica a perossido di idrogeno: viene usato il macchinario STERRAD che garantisce un tempo di sterilizzazione pari a 45 minuti e kit sterili fino a 6 mesi.

Lo strumentario viene riposto all'interno del macchinario in appositi sacchetti con un bollino giallo, che indica la contaminazione del materiale riposto all'interno.

Dopo l'avvenuta sterilizzazione la macchina smette di lavorare, il sacchetto viene spostato su un tavolo, con il bollino che all'interno del macchinario si è colorato di verde, e viene apposta un'etichetta con data e ora dell'avvenuta sterilizzazione per poi essere riposto in appositi scaffali.

L'ospedale Stecca Bianca ha affidato le operazioni di decontaminazione, riordino e ripristino delle sale, trasporto dei pazienti e lavaggio manuale dei ferri a un'azienda ausiliaria.

La decontaminazione è un tema molto delicato in ambito sanitario per cui è necessario seguire protocolli ben definiti sia per quanto riguarda i flussi del personale da un ambiente a un altro, sia per quanto riguarda la disinfezione dello strumentario e delle sale operatorie.

Il paziente da operare in regime di elezione accede al Blocco Operatorio mediante un Passamalati, un letto sterilizzato, al fine abbattere la carica batterica che si avrebbe se l'ingresso avvenisse su una barella.

Qualora il paziente, invece, debba essere operato in urgenza accede sulla barella da una porta a due ante secondaria.

Gli operatori del comparto accedono al blocco dopo essersi cambiati e aver indossato indumenti sterili.

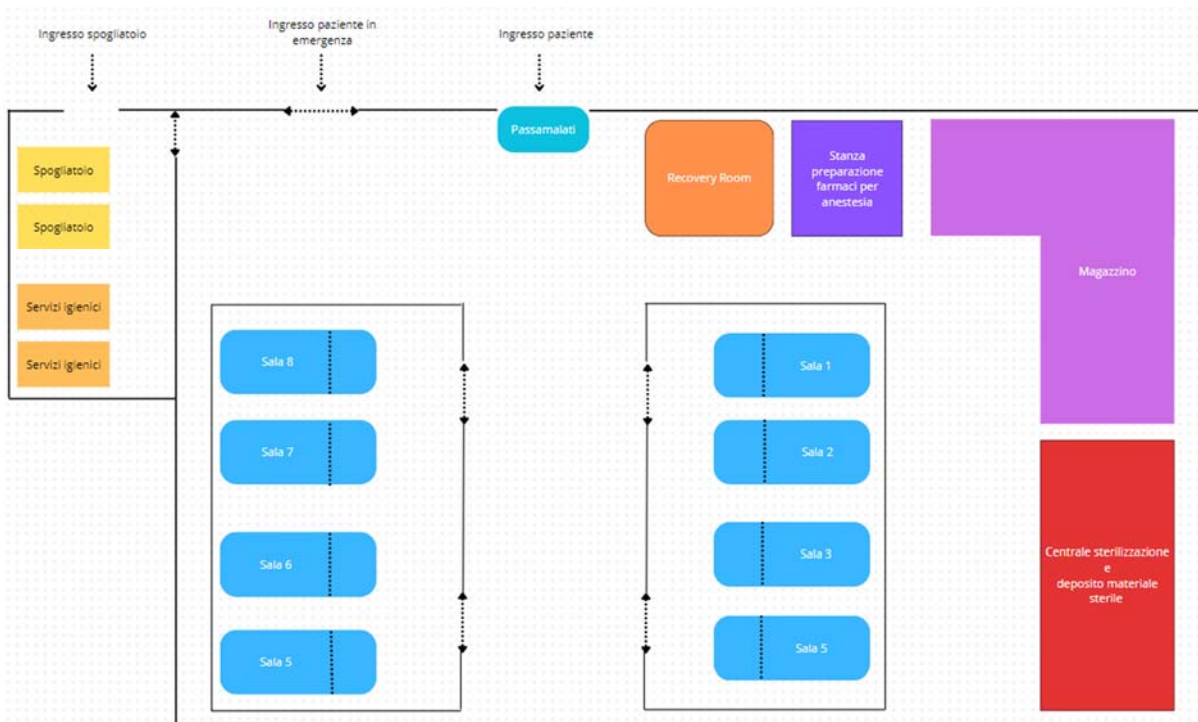


Figura 2: Schema blocco operatorio della Stecca Bianca - Canva

La Recovery Room, o Sala Risveglio, permette la gestione del paziente al suo risveglio attraverso l'assistenza di personale specializzato, pronto a intervenire in caso di complicanze o di effetti residui dei farmaci somministrati.

È stato stimato che una grossa fetta di eventi avversi nel post-operatorio si verifica nelle ore immediatamente successive alla dimissione del paziente dalla sala operatoria.

La dotazione delle recovery room è definita dalla Direzione Sanitaria e prevede che ogni postazione sia attrezzata sulla base della tipologia degli interventi previsti per la giornata.

2.2. Le liste d'attesa

In Italia, la domanda delle prestazioni sanitarie rivolte al settore pubblico è considerata anelastica, ossia la quantità domandata non è sensibile alle variazioni di prezzo perché le spese relative alle prestazioni offerte sono a titolo gratuito o, al massimo, fornite dietro pagamento di un ticket.

Le cure mediche sono beni essenziali, motivo per cui la domanda non è influenzata dalla variazione dei costi, che restano in capo al Sistema Sanitario Nazionale Italiano [2][17].

I lunghissimi tempi di attesa e la qualità del servizio, talvolta non adeguata, potrebbero far optare al paziente l'utilizzo del settore sanitario privato al fine di ottenere un'assistenza

superiore in termini di servizio offerto.

Questo è un problema ricorrente nei servizi pubblici per quanto assolutamente risolvibile.

Le aziende private ragionano per target ed obiettivi, al fine di offrire la prestazione migliore possibile in tempi definiti.

Imitare questo paradigma nei servizi pubblici sembra impossibile.

Per importarlo, adattarlo ed implementarlo nella sanità è necessario che i ruoli all'interno di questa complessissima struttura gerarchica siano ben definiti e, con essi, anche le responsabilità di ogni membro.

Uno dei problemi più evidenti, riscontrato in prima persona, è infatti la responsabilità condivisa, alla quale segue una caccia all'errore e successivamente una caccia al colpevole con scarsissimi risultati. Avere dei ruoli definiti e delle responsabilità chiare, infatti, permetterebbe uno schema analogo al settore privato: raggiungimento di obiettivi prestabiliti e applicazione di sanzioni qualora questi non fossero rispettati, il tutto seguito da operazioni di controllo periodiche da parte di operatori esterni alla stessa struttura.

Questo tipo di operazioni renderebbe il sistema sanitario italiano realmente accessibile a tutti, senza venir meno ai principi di Universalità (accesso a prestazioni sanitarie a tutta la popolazione), Uguaglianza (accesso senza alcuna distinzione sociale) ed Equità (parità di accesso per tutti di fronte alla parità di bisogni), Solidarietà, Centralità del Paziente e Gratuità o quote di partecipazione limitate [2].

Le liste d'attesa presentano un altro problema dal punto di vista operativo: i pazienti non sempre vengono registrati correttamente, in quanto nell'ambiente sanitario non è necessariamente compresa l'importanza dell'utilizzo e dell'estrazione di dati puntuali.

L'*equipe* chirurgica è tenuta ad effettuare la procedura chirurgica e a registrare i dati inerenti agli orari di entrata, anestesia e uscita del paziente dalla sala e, soprattutto, al nosologico (inserimento univoco del paziente) e alla cancellazione del paziente dalla lista.

Se questo *modus operandi*, da un lato, non permette una corretta estrazione del dato e conseguente programmazione non attendibile, dall'altro crea disagio sanitario anche ai pazienti.

La resistenza al cambiamento nel passaggio da cartelle cartacee a quelle virtuali è evidente. L'utilizzo di servizi sanitari digitalizzati potrebbe facilitare lo scambio di informazioni sanitarie sui pazienti e sull'affollamento delle strutture, permettendo, per esempio, al paziente di poter scegliere tra un Pronto Soccorso e un altro.

2.3. I posti di terapia intensiva post-operatoria

L'intervento chirurgico e le patologie pregresse del paziente sono le due condizioni che consentono al medico di stabilire anticipatamente se per il paziente sarà necessario l'utilizzo della terapia intensiva post-operatoria.

La terapia intensiva post-operatoria viene spesso nominata con l'acronimo TIPO [18].

In linea generale, interventi chirurgici più complessi o invasivi possono necessitare una terapia intensiva post-operatoria.

Inevitabilmente alcune specialità chirurgiche, come Chirurgia Vascolare, sono più soggette all'utilizzo dei posti in terapia intensiva post-operatoria rispetto ad altre.

Questo non esclude, però, che anche specialità come la Terapia Antalgica, che si occupa prevalentemente della gestione del dolore cronico o acuto, possa, eccezionalmente, farne richiesta. La TIPO può essere utilizzata anche da pazienti che richiedono interventi comuni: una frattura di femore di un degente al quale anni prima è stata sostituita una valvola aortica, ad esempio, può essere valutato come intervento ad alto rischio.

Il controllo intensivo post-operatorio richiede un rapporto infermiere-paziente 1:1, sono dunque molto dispendiosi in termini di costi e risorse [19].

I posti in TIPO sono una risorsa scarsa, sia per il numero esiguo di posti letti dedicati, sia per la necessità della supervisione di infermieri che, a loro volta, sono una risorsa scarsa.

Questi posti letto vanno differenziati dai posti in terapia intensiva.

Per quanto concerne l'ospedale di Sassari del Blocco Operatorio del Plesso di San Pietro, i posti in terapia intensiva sono 23. Due di questi sono dedicati alla terapia intensiva post-operatoria e uno alle EU (Reparti di Emergenza/Urgenza).

I due posti vengono occupati con gli interventi che prevedono, già dalla schedulazione giornaliera, l'utilizzo dei posti in TIPO, mentre il terzo rappresenta il posto libero che deve essere lasciato tale, per affrontare:

- pazienti che presentano complicanze durante l'intervento
- altre urgenze.

Ulteriori EU vengono regolarmente gestite dalla terapia intensiva (non post-operatoria): i pazienti vengono accolti e gestiti eccezionalmente dalle risorse presenti in TIPO.

L'*iter* operativo di un paziente in terapia intensiva post-operatoria è il seguente: uscita dalla sala operatoria, ingresso in terapia intensiva post-operatoria e, dopo un tempo t , spostamento nel reparto o, eventualmente, nella terapia intensiva.

È stimata una giacenza media nel singolo letto di 48 ore, indipendentemente dall'operazione.

Ad oggi, il Direttore della S.S.D. Operation Manager and Operating Room Manager organizza una volta alla settimana (di giovedì) una riunione alla quale devono partecipare la Dirigente medico Anestesia e Rianimazione, il Referente Clinico Terapia Intensiva Post-Operatoria e tutti i Direttori o referenti di tutte le specialità.

La riunione si articola con la presentazione, da parte dei referenti di ciascuna specialità chirurgica, dei propri casi TIPO e la successiva discussione con tutti i Direttori al fine di dare una priorità a questi pazienti.

Si pianifica l'intervento di due pazienti al giorno: il primo con assoluta priorità e un secondo, da immaginare come riserva, che il più delle volte non viene operato.

Il secondo paziente della lista spesso presenta meno comorbidità di quello schedato come primo paziente di ogni giorno.

Il giorno seguente si continua a seguire il programma, dunque non si dà posto necessariamente al secondo paziente TIPO non operato il giorno prima.

Prima di ogni intervento che preveda la TIPO, è necessario chiamare il responsabile della terapia intensiva, al fine di valutare lo stato della stessa, il numero di letti e di risorse disponibili onde poter valutare se sia il caso di accogliere l'arrivo in giornata del paziente.

Avere un paziente in terapia intensiva post-operatoria implica avere molte responsabilità e avere una risorsa dedicata al paziente, per cui sovente accade che a priori venga data risposta negativa.

Spesso i posti in TIPO vengono occupati irregolarmente e, a causa delle esigue risorse infermieristiche, per la Terapia Intensiva, l'arrivo di un nuovo paziente rappresenta tipicamente un incremento di criticità andando ad occupare al 100% una risorsa infermieristica.

Bisogna quindi dare massima priorità ai posti letto della TIPO.

Ottimizzando questi posti letto, ossia prevedendo che i posti in terapia intensiva post-operatoria siano sempre a pieno regime, si ottiene che, i posti letto occupati irregolarmente vengano liberati, con un conseguente abbattimento sostanzioso delle liste d'attesa ma soprattutto facendo sì che questi pazienti possano avere un'assistenza adeguata alla loro condizione sanitaria.

L'intervento in TIPO è un intervento ad alta complessità, dunque è un intervento che presumibilmente occuperà l'intera sala per tutto il giorno.

Operando prima questa categoria di pazienti, bisogna prevedere, dunque, che due sale sulle otto disponibili siano quotidianamente dedicate ad essi.

Corrispondentemente, allo scopo di trattare tutte le specialità in modo equivalente, si è deciso di far ruotare le specialità per la TIPO e si è, quindi, ipotizzato in questa tesi, che le specialità

scelte per la TIPO, qualora abbiano un paziente, ruotino settimanalmente.

Lavorando 5 giorni a settimana potranno, quindi, essere operati 10 pazienti TIPO di 10 specialità diverse in ordine di arrivo.

Questa assunzione prende ovviamente in considerazione un lavoro a pieno regime, al netto di festività. L'ordine di arrivo è gestito dall'IDPrenotazione, un numero progressivo che viene assegnato al paziente nel momento in cui viene immesso nella lista d'attesa.

È un numero di 16 cifre che aumenta progressivamente, senza tener conto della specialità.

Il motivo per cui si è deciso di scegliere “il primo della lista d'attesa con specialità differente” al posto di fare un check sulla priorità è che un paziente TIPO è sempre un paziente di priorità A, per cui, a parità di priorità, si sceglierà il primo arrivato.

L'attributo che differenzia il paziente TIPO dal paziente non-TIPO è una variabile booleana, data in input dal Chirurgo dello stesso paziente dopo aver valutato la storia clinica del paziente e la sua condizione attuale.

2.4. Le risorse infermieristiche

Il personale infermieristico lavora per un totale di 36 ore settimanali. Generalmente lavora 5 giorni a settimana, dunque, opera su turni giornalieri di 7 ore e 12 minuti.

Il gruppo di lavoro che opera nel primo slot entra a lavoro alle 7 e finisce alle 14:12, il secondo dalle 12:48 alle 20.

La schedulazione degli slot operatori viene fatta sempre su due turni: il turno del mattino ha inizio alle 8 per le 6 ore successive fino alle 14 e il turno del pomeriggio, che inizia alle 14 e finisce alle ore 20.

Il tempo necessario per il cambio degli indumenti deve essere escluso dal tempo del turno infermieristico, al contrario di eventuali pause che vengono concesse, ove necessarie.

Dopo aver ridotto la loro carica batterica, cioè solo dopo essersi cambiati, gli infermieri possono e devono timbrare entro l'orario prestabilito, salvo esigenze approvate dal loro coordinatore.

Il tesserino di ciascun infermiere deve essere dunque riposto in uno spazio adibito internamente al blocco operatorio, non all'esterno, onde evitare eventuali scorrettezze.

Un altro punto di attenzione è relativo alla gestione di ferie e permessi per motivi vari.

Il processo di assegnazione delle ferie dovrebbe essere un processo rolling che consiste in una pianificazione a monte, successivamente modificata sulla base delle esigenze emerse a valle.

È un processo di pianificazione che prevede la schedulazione su un arco di tempo che va da un mese a un trimestre.

A inizio anno, la Direzione Generale, con la collaborazione della Direzione Strategica, decide gli obiettivi strategici e, sulla base di essi, calcola il fabbisogno di organico medio, al fine di aprire il numero di sale previste.

Il Coordinatore infermieristico, dopo aver visionato la proposta della Direzione Strategica, deve organizzare un piano ferie e valutare la fattibilità della proposta a monte.

In un primo momento devono essere allocate le ferie obbligatorie: due settimane di ferie estive, garantite a tutti gli operatori, e 15 giorni consecutivi, non interrompibili, di riposo biologico per esposizione ai raggi X, per i dipendenti soggetti a tale esposizione.

Dopo aver allocato queste ferie, si distribuisce nei mesi restanti la restante quota-parte delle ferie, maggiorata di una percentuale [19][20].

La percentuale considerata permette di prevedere un numero di assenze medie, stimate sul periodo, al fine di far avvicinare sempre più il programmato al consuntivo.

A questo punto, il Coordinatore degli infermieri deve comunicare e proporre alla Direzione Strategica eventuali modifiche al piano proposto.

Le ferie vanno distribuite su tutto l'anno, al fine di limitare l'allontanamento dal Rough Cut Capacity Planning (RCCP).

L'RCCP verifica se la capacità produttiva disponibile in un'azienda sia sufficiente a soddisfare i requisiti della stessa in termini di capacità produttiva richiesta dai principali programmi [21].

Qualora sopraggiungano nuove richieste di assenze a medio lungo termine per esigenze sopravvenute o per malattia, il coordinatore degli infermieri deve contattare la Direzione al fine di verificare se redistribuire le ferie rimanenti sul periodo dell'anno restante o se decidere di concentrarle in quel periodo, riducendo così l'attività chirurgica dell'intervallo.

Una delle proposte di miglioramento e di supporto al processo è stata quella di informatizzare le utenze infermieristiche e suddividere i giorni di ferie, al fine di preventivarli in modo oggettivo e, solo successivamente, decidere quali, in quale periodo e quante risorse extra possano assentarsi per permessi.

3. La schedulazione delle sale operatorie

3.1. Revisione della letteratura

La schedulazione delle sale operatorie è oggetto di studio da svariati decenni [27].

Si vuole trovare un trade off tra l'ottimizzazione dei processi e il miglioramento della gestione delle risorse e l'usabilità del sistema da parte degli operatori preposti.

Alla base di tali sistemi che sfruttano tecnologie avanzate, vengono utilizzati algoritmi complessi che vengono spesso importati da altre realtà e riadattati all'utilizzo dei servizi sanitari.

La differenza tra altre realtà e quella sanitaria è inevitabilmente il lato etico-umano, spesso trascurato al fine di massimizzare la gestione ottimale delle risorse, motivo per cui vengono difficilmente accettati dagli operatori sanitari [35][36].

Affinché un sistema informatico venga preso in considerazione in questo settore, deve avere un'elevata usabilità e se ne deve far comprendere l'importanza in termini di riduzione del carico di lavoro infermieristico, di riduzione degli sprechi e di oggettività delle decisioni prese, consentendo una riduzione drastica della discrezionalità nello stilare la lista operatoria [22][38].

Alcuni dei vantaggi nella schedulazione delle sale operatorie includono:

- Pianificazione smart: la pianificazione, una volta implementati i dati in input, avviene pressoché real time, senza l'utilizzo di personale preposto;
- Gestione ottimale delle sale operatorie e delle ferie del comparto: schedulando le diverse specialità nelle sale, si riduce la possibilità di ferie indesiderate da parte del comparto che, se necessario al programma, una volta stilato quest'ultimo, non può assentarsi salvo, in casi di malattia. La pianificazione, inoltre, generalmente lavora leggermente in overbooking, favorendo l'inserimento di pazienti extra in una delle eventuali sale libere [39][40].
- Gestione delle risorse: una pianificazione adeguata degli interventi consente di organizzarsi in termini di programmazione delle ferie, di tutela del paziente e di riordino delle materie prime. Lo stoccaggio di materiale negli ospedali è un problema da non sottovalutare in quanto accumuli di scorte fanno aumentare il capitale immobilizzato con rischio di deterioramento o di inutilizzo in caso di superamento della data di scadenza. Considerando che il materiale utilizzato in ospedale ha costi

non irrilevanti, una pianificazione accurata degli interventi consente una riduzione sostanziale di tali fattori [37][40].

- Un'accurata pianificazione delle procedure aiuta a ridurre le cancellazioni dell'ultimo minuto, che possono essere costose e avere un impatto negativo sia sui pazienti che sul personale medico.
- Tracciamento: l'informatizzazione consente il tracciamento e monitoraggio del dato in tempo reale [40].

3.2. Osservazioni e criticità dell'attuale gestione

Alta discrezionalità, limitata organizzazione delle risorse umane e non, responsabilità condivisa e scarsa cultura del dato sono l'attuale paradigma ospedaliero.

Attualmente l'allocazione delle specialità chirurgiche nelle diverse sale e slot operatori viene decisa e inviata dallo staff dell'Operating Room Manager al blocco operatorio alla fine del mese precedente per il mese successivo.

L'unica programmazione che viene fatta su un orizzonte superiore a un mese è quella dei mesi estivi: si programmano le sale operatorie elettive da giugno a settembre per via delle ferie/assenze.

La schedulazione viene fatta a tavolino, sulla base delle ore settimanali di sala che spettano ad ogni specialità chirurgica.

Ogni reparto considerato nella schedulazione della giornata invia la lista operatoria del giorno successivo alla mail del blocco operatorio.

Questa operazione viene fatta solo il giorno prima per il giorno successivo.

Teoricamente la deadline è prevista alle 12 del giorno precedente ma, nella realtà, non essendoci sanzioni, spesso e volentieri la scadenza non viene rispettata.

Nonostante il poco preavviso, è stata stimata una media del 10% di rettifiche di lista che comprende modifiche fatte la notte per la mattina e la mattina per la mattina.

Per alcune specialità si raggiunge settimanalmente anche il 40% delle modifiche.

Si può affermare che la programmazione ottenuta dal percorso sopra descritto, nonostante venga realizzata oggi per domani, è anche non troppo attendibile.

La mail del blocco operatorio viene letta sia dell'Operating Room Manager sia dal coordinatore degli infermieri.

Quest'ultimo stampa la lista operatoria giornaliera, informa le risorse del blocco operatorio

che, solo a quel punto, possono e devono prepararsi agli interventi del giorno dopo in meno di 20 ore.

Per quanto riguarda i pazienti che necessitano di maggiore assistenza post-operatoria, viene istituita una riunione il giovedì per la settimana successiva. In particolare, non c'è alcuna regola che indichi quando operare pazienti di priorità inferiori alla A per cui anche questo è a discrezione del chirurgo di riferimento che può decidere se dedicare un'intera giornata ogni tot settimane a pazienti di priorità diversa da A o se inserirli contestualmente a pazienti di priorità A senza seguire necessariamente una logica precisa.

Il poco preavviso riguardante l'allocazione delle specialità chirurgiche nella settimana e le decisioni riguardanti quali pazienti e in quale sala operarli crea in generale malfunzionamenti e disorganizzazione.

Sapere che tipo di interventi si dovranno sostenere nel lungo termine consentirebbe di avere benefici, lato ospedale e lato paziente.

Dal punto di vista ospedaliero si riuscirebbe, così a:

- avere il tempo per preparare l'equipe di chirurghi, infermieri e tutto ciò che occorre per gli interventi;
- consentire di avere una sala adeguatamente allestita dal giorno precedente,
- evitare sprechi di tempo, riducendo il rischio di far slittare pazienti alla seduta successiva;
- acquistare materiali in quantità coerente con gli interventi schedulati riducendone corrispondentemente i costi;
- ridurre rischi di scadenza e deterioramento dei materiali, dovuto a un eccesso di scorte inutilizzate.

Una data d'intervento prevista affidabile consentirebbe al paziente di prepararsi all'intervento, avendo il tempo di organizzarsi ove dovesse adempiere a esigenze lavorative e familiari riducendo il rischio di dover ripetere il medesimo iter a causa dello slittare dell'intervento stesso.

Alla luce di ciò è ragionevole affermare che programmare le sale operatorie con alcune settimane di anticipo consente di attutire l'impatto e, talvolta, evitare gran parte di questi problemi. La schedulazione delle sale e degli interventi consente inoltre di effettuare un adeguato piano di mitigazione dei rischi.

3.3. Un approccio costruttivo scritto in codice C++

Un primo approccio di tipo costruttivo scritto in codice C++ viene presentato qui di seguito. Tale approccio prevede di considerare gli interventi in modo progressivo in funzione dell'anzianità di inserimento in lista nel rispetto dei vincoli di priorità tra le classi A, B, C, e D. Il programma permette di visualizzare in output il piano operatorio nel breve e nel lungo periodo. È possibile, infatti, ottenere una schedulazione delle sale dal primo giorno disponibile fino ad un massimo di 52 settimane con l'indicazione di quali pazienti verranno operati (con la relativa specialità chirurgica cui facciano riferimento) e quando lo debbano essere, seguendo criteri oggettivi. Ai chirurghi si richiede di implementare la scheda del paziente con un flag, qualora necessiti di accedere alla Terapia Intensiva Post-Operatoria e un codice "durata prevista intervento".

Il secondo dato non prevede una durata precisa, ma semplicemente una stima secondo un sistema di codifica che divide gli interventi in seduta operatoria corta, qualora siano interventi in anestesia locale, media, lunga o che richiedono la sala per l'intera giornata.

L'approccio costruttivo prende in considerazione otto sale operatorie, interscambiabili, consentendo l'efficientamento dei processi del blocco operatorio.

Tale approccio deve essere visto come uno strumento di supporto, utile a ridurre sostanzialmente il carico di lavoro del Coordinatore infermieristico, che si occupa dell'inserimento dei pazienti da operare per ogni specialità.

La flessibilità dell'output ottenuto prende, però, in considerazione l'eventuale necessità di modifiche dell'ultima ora e conseguentemente permette di aggiornare opportunamente la lista operatoria.

Il programma, dopo essere stato scaricato può essere implementato con variazioni e modifiche non necessariamente prevedibili a priori.

Tra queste, vengono considerati quei pazienti che presentano problemi di varia natura pre-intervento e che non possono essere operati per motivi di salute sopravvenuti a ridosso dell'intervento, oppure interventi non più effettuabili per decesso del pazienti, oppure pazienti che si aggravano o per i quali sopraggiungono nuove comorbidità, per cui viene modificata la priorità chirurgica o più semplicemente la presenza di interventi la cui durata sia significativamente superiore o inferiore alla durata prevista.

Per quanto riguarda la gestione delle problematiche riguardanti il paziente, dunque decessi, comorbidità sopraggiunte in un secondo momento o lo slittare dell'intervento per altri motivi, la gestione prevista è piuttosto simile.

Nei primi due casi, si interviene nel file *Lista d'attesa* in input rimuovendo il paziente in caso di decesso e cambiando la priorità al paziente in caso di nuove disposizioni da parte del medico chirurgo.

Seppur i tempi stimati per gli interventi siano tipicamente piuttosto attendibili, devono essere considerati e gestiti i casi di tempo in esubero nella sala di riferimento oppure in difetto qualora non si riesca ad operare tutti i pazienti schedulati.

Nel primo caso si aggiungono manualmente uno o più pazienti nella programmazione operatoria, seguendo i medesimi criteri stabiliti e utilizzati per gli altri pazienti, e successivamente li si rimuove manualmente dalla lista d'attesa.

Nel secondo caso seguente, invece, si fanno slittare gli interventi dei pazienti non operati ai primi slot di tempo disponibili.

Una delle ipotesi presa in considerazione è di lavoro a pieno regime, dunque su tutte le otto sale, senza tener conto della disponibilità del personale infermieristico.

Tale opzione, però, può causare criticità essendo il personale infermieristico una risorsa scarsa ed il numero di infermieri da coinvolgere nell'apertura delle sale spesso insufficiente. Qualora si vogliano o debbano aprire meno sale per questa ragione, si deve modificare il codice, segnando il reale numero di sale apribili in quella determinata giornata, al netto degli infermieri disponibili.

Gli interventi oggetto del programma sono i cosiddetti interventi in elezione. Si tratta di interventi previsti solo nei giorni feriali, quindi non nel fine settimana né nelle giornate festive sancite a livello nazionale, regionale o patronali del Comune di riferimento. In tali giornate è prevista attività chirurgica solo in regime di emergenza e urgenza.

Per realizzare un algoritmo solido è necessario stabilire criteri e regole ben strutturate a monte che riescano a prevedere quante più eventualità, anche le meno probabili, dando poco spazio alla discrezionalità.

Questo codice è stato scritto con questi presupposti.

L'obiettivo del programma è quello di massimizzare l'utilizzo dei posti in Terapia Intensiva Post-Operatoria (TIPO, risorsa scarsa) e, successivamente, l'ottimizzazione delle altre sale operatorie, grazie a una schedulazione dei restanti slot operatori elettivi sulla base di criteri prestabiliti.

Prevedendo operazioni chirurgiche per 5 giorni a settimana, si raggiunge l'ottimo di interventi di pazienti che necessiteranno della TIPO, se il loro numero sia pari a n. 10 pazienti a settimana.

Ove, però, la settimana comprenda giornate festive, l'ottimo si raggiungerà per un numero di

interventi inferiore.

DATI DI INPUT

Il programma, per poter funzionare correttamente, deve poter attingere a tre file txt in input denominati *GO_calendario2024*, *GO_listaattesa* e *GO_numLettiPerSpecialita* con il formato sotto descritto.

I file utilizzati per testare il programma sono sulla falsariga di quelli utilizzati realmente dagli ospedali.

La struttura è la seguente:

GO_calendario2024: Data, #ggsettimana, festivo(1)/lavorativo(0)

- La data è scritta nel formato aaaa-mm-dd
- Il giorno della settimana è un numero intero numerato da 1 a 7; il lunedì è associato al numero 1, il martedì al numero 2 e così via.
- Il dato festivo/lavorativo è una variabile booleana:
 - Sono contrassegnati con 1 i giorni di festività Nazionale, i sabati e le domeniche. In questi giorni non sono previsti interventi in elezione che sono presenti 5gg/7gg in una settimana priva di festività;
 - i giorni lavorativi sono contrassegnati con 0.

In questo caso, il calendario utilizzato per testare la validità del programma è del 2024.

Tiene conto delle festività nazionali, dei sabati e delle domeniche ma non delle festività regionali e comunali.

È possibile, inoltre, modificare il contenuto del file aggiungendo tante righe quante desiderate per cui non è vincolante la decisione di utilizzare un orizzonte temporale prefissato.

GO_listaattesa: IdPaziente, Specialità Chirurgica, Priorità Clinica, TIPO(1)/NonTIPO(0), Lunghezza Intervento

- L'IdPaziente è un numero che viene assegnato al paziente nel momento in cui viene inserito nella lista d'attesa. È un numero progressivo, di 16 cifre, che aumenta a ogni inserimento di nuovi pazienti, indipendentemente dalla specialità chirurgica.

- La Specialità Chirurgica è un'abbreviazione del nome della specialità, è un char. Ci sono 16 specialità chirurgiche.
- Priorità Clinica: è un carattere, può essere A, B, C o D.
- TIPO/NonTIPO: è una variabile booleana: è pari a 1 se il paziente necessita della TIPO, 0 viceversa.
- Lunghezza intervento, è un char:
 - C: intervento di corta durata, si possono operare 4 pazienti al giorno per sala;
 - L: intervento di lunga durata, si possono operare 2 pazienti al giorno per sala;
 - E: intervento in anestesia locale, si possono operare 8 pazienti al giorno per sala;
 - T: intervento di un paziente che necessita la terapia intensiva post-operatoria, si può operare 1 solo paziente al giorno per sala.

GO_numLettiPerSpecialita: Nome specialità, Quantità di slot operatori dedicabile.

- Nome specialità: è un'abbreviazione coerente con il nome definito prima in specialità chirurgica.
- Quantità di slot operatori dedicabile è un intero. Questo numero rappresenta il peso di una determinata specialità chirurgica sul totale in relazione al numero di pazienti che le compete. Aiuta a definire quanti letti e di conseguenza slot operatori destinare ad una specialità chirurgica piuttosto che ad un'altra.

DATI IN OUTPUT

Una volta lanciato, il programma consente di visualizzare il seguente menù, guidando l'utente nell'utilizzo del programma:

Digita uno dei seguenti numeri per selezionare una delle opzioni desiderate:

- 1) Visualizzare la programmazione operatoria in data odierna;*
- 2) Passa al giorno seguente;*
- 3) Visualizzare la programmazione operatoria nella settimana corrente;*
- 4) Exit".*

Opzione scelta:

È ragionevole attribuire il ruolo di utente al coordinatore degli infermieri.

L'utente sceglie una delle seguenti opzioni ed eventualmente implementa la programmazione quotidianamente sulla base delle eventualità sopraggiunte nel momento,

come descritto precedentemente.

Il file di output è organizzato al fine di avere dei files .txt separati per ogni settimana.

Il coordinatore può decidere di richiedere la programmazione odierna giorno per giorno senza però riempire la cartella di output con un numero eccessivo di files, di difficile lettura.

L'utente, dunque, può visualizzare la programmazione in data odierna, il giorno seguente chiedere di passare al giorno seguente e successivamente richiedere la programmazione della data odierna e così via: il lunedì di quella settimana si crea un file di output intitolato con gli estremi di quella settimana, si riempie con i dati di quel giorno, poi del giorno seguente e così via.

Quando si passa al lunedì di una nuova settimana, viene creato un nuovo file che verrà riempito con i dati di quella settimana. I files, settimana dopo settimana vengono salvati e riempiono la cartella di output.

L'iter da seguire è più snello del precedente.

Il coordinatore degli infermieri deve procedere all'estrazione della programmazione quotidiana o settimanale, ad apportare eventuali operazioni di modifica nel file di output al fine di completare la programmazione e tener traccia della stessa per ogni evenienza.

Successivamente deve inviare una mail con in allegato la programmazione settimanale all'ufficio di Operations Manager che procede ad effettuare un lavoro di monitoraggio e controllo sulla schedulazione al fine di verificare e segnalare eventuali illeciti o irregolarità dell'operato settimanale.

Ovviamente, il file di input utilizzato, scorrendo il calendario del 2024, parte dal 1° gennaio 2024 e prende in considerazione le festività Nazionali, i sabati e le domeniche del 2024.

CONTENUTO

Gestione pazienti che necessitano di terapia intensiva post-operatoria.

La gestione dei pazienti TIPO è strettamente connessa a quelli non-TIPO seppur segua regole distinte.

Le due categorie di pazienti comunicano ma vengono gestite in due blocchi differenti.

Inizialmente si ipotizza di avere tutte le sale operatorie vuote e, volendo massimizzare le operazioni della categoria di pazienti TIPO, si scheduleranno questi per primi.

I letti in Terapia Intensiva Post-Operatoria sono da considerare una risorsa scarsa in quanto sono solo due al giorno.

Questi pazienti hanno priorità A di default per via dei rischi legati alla natura dell'intervento

o a causa delle comorbidità del paziente stesso.

A causa della delicatezza e della complessità di questi interventi, la durata dell'intervento è da considerarsi lunga, dunque, un paziente occupa una sala per l'intera giornata lavorativa.

Dopo aver ordinato la lista d'attesa in ordine di arrivo dei pazienti, ossia in ordine di IdPrenotazione, si sceglie il primo paziente TIPO da operare e si segna la sua specialità chirurgica in una tabella.

Si impone che il secondo paziente operato lo stesso giorno abbia una specialità diversa da quella del paziente operato precedentemente.

Questo vincolo è stabilito dal fatto che l'equipe chirurgica è una per specialità e dunque, se impegnata in un paziente che necessita la TIPO, è da considerarsi già occupata.

Successivamente, al fine di gestire in modo uniforme tutte le specialità chirurgiche, ossia al fine di evitare che vengano operati solo o prevalentemente pazienti di una specialità a scapito di altre, si fanno ruotare le specialità chirurgiche di pazienti TIPO operati, fino all'esaurirsi di tutte le specialità.

Qualora ci fossero una o più specialità chirurgiche che non prevedano pazienti TIPO o le specialità chirurgiche siano esaurite, il programma:

- “svuota la tabella” nella quale si teneva traccia delle specialità dei pazienti operati in TIPO;
- ricomincia programmando i pazienti dalla prima specialità interessata ossia dal primo paziente in lista che necessita questo trattamento indipendentemente dalla specialità. Non è necessario dunque seguire l'ordine utilizzato la settimana precedente, ogni settimana è indipendente dall'altra. Dopo aver preso il paziente dalla lista si segue l'iter descritto precedentemente: si verifica che non abbia la stessa specialità degli altri pazienti schedulati né in TIPO né nelle altre sale operatorie lo stesso giorno. Successivamente si segna la specialità trattata nella tabella.

Dopo essere stato operato, il paziente viene rimosso dalla lista d'attesa.

Riassumendo, i vincoli utilizzati per la schedulazione dei pazienti che necessitano la TIPO sono:

- Operazioni effettuate per 5 giorni alla settimana (i sabati e le domeniche sono considerati festivi). Si escludono dalle giornate lavorative anche le altre festività Nazionali.
- Rotazione delle specialità fino ad esaurimento delle stesse.

- Equipe: se l'equipe è impegnata nell'intervento del paziente 1-TIPO, non può impegnarsi nell'intervento del paziente 2-TIPO né tantomeno nell'intervento di altri pazienti non TIPO in altre sale operatorie, perché già occupata.
- Letti: si hanno due soli letti di terapia intensiva post-operatoria destinati, per cui si possono operare al più due pazienti ritenuti critici al giorno.
- Utilizzo della sala per l'intera giornata lavorativa.

Dopo aver schedato e ottimizzato questa categoria di pazienti, il codice schedula i restanti slot operatori elettivi.

Il totale delle sale è otto e, considerando che due di esse sono occupate per pazienti che necessitano la TIPO, si hanno sei sale operatorie restanti da ottimizzare.

Gestione pazienti che NON necessitano di terapia intensiva post-operatoria.

PRIORITA' E SALE DEDICATE

È necessario distinguere i pazienti di priorità diverse.

I pazienti possono avere priorità A, B, C e D. I pazienti TIPO, facenti parte della priorità A, sono ovviamente scartati da questa gestione.

Si dedicano quattro delle sei sale restanti, a fronte delle otto totali, ai pazienti di priorità A, mentre le restanti due ai pazienti di priorità B, C e D.

Queste attività di schedulazione sono consequenziali, dunque non avvengono in parallelo, in quanto si schedulano inizialmente i pazienti di priorità A e, solo successivamente, si procede all'assegnazione degli slot operatori agli altri con priorità meno impellenti.

Qualora venissero operati tutti i pazienti di priorità A, il programma attinge ai pazienti di priorità B, C e D in questo ordine, al fine di riempire la programmazione settimanale.

Solo dopo aver segnato in una nuova tabella le specialità dei pazienti operati nelle 4 sale dedicate alla priorità di tipo A, si procede all'assegnazione degli stessi alle sale dedicate a interventi con priorità inferiore che, secondo gli stessi criteri, devono subire un'operazione appartenente a specialità, assegnate a rotazione, differenti da quelle che operano nelle altre sale.

Per quanto concerne i pazienti di priorità diversa da A, si alternano le due sale rimaste assegnandole, ogni settimana, due giorni ai pazienti di priorità B, due giorni ai pazienti di priorità C e un solo giorno ai pazienti di priorità D.

Qualora ci siano uno o più festività nazionali nella settimana di riferimento, si continua a operare con questo ordine, dunque, si inizierà sempre ad assegnare slot ai pazienti di priorità

B.

Qualora tutti i pazienti B siano stati operati, si destinano gli slot operatori dedicati a questi ultimi, ai pazienti di priorità C e D, in questo ordine. Terminati anche questi, le sale saranno riassegnate ai pazienti con priorità A.

Una volta allestita per una determinata specialità, la sala deve continuare ad operare pazienti di quella determinata specialità.

DURATA INTERVENTO

È necessario inoltre porre attenzione alle durate degli interventi in modo da allocare i pazienti nelle sale nel miglior modo possibile sulla base della durata degli interventi.

Come detto precedentemente, questi dati, sono reperibili dal file lista d'attesa:

- C: intervento di corta durata, si possono operare 4 pazienti al giorno per sala;
- L: intervento di lunga durata, si possono operare 2 pazienti al giorno per sala;
- E: intervento in anestesia locale, si possono operare 8 pazienti al giorno per sala;
- T: intervento di un paziente in tipo, si possono operare 1 solo paziente al giorno per sala (ora non ci interessa più).

Si immagina la giornata lavorativa come se fosse costituita da 8 slot operatori: un paziente di durata E ne occupa 1, uno C ne occupa 2, uno L ne occupa 4 e uno T 8.

Dati questi vincoli, si riesce a incastrare pazienti con interventi di durata E con pazienti di durata C ed L e viceversa.

SCELTA DEL PAZIENTE

Si ha una specialità diversa in ogni sala operatoria in quanto, presso l'Ospedale Stecca Bianca di Sassari, è prevista una sola equipe chirurgica per ogni specialità in un dato slot.

PER OGNI SALA

- Si sceglie la specialità da collocare nelle varie sale facendo attenzione al peso che una determinata specialità chirurgica ha sul totale degli slot. È un dato predefinito, che si trova nel file contenente i dati di ciascuna specialità e che si azzerà settimanalmente. Permette di definire quanti slot operatori (chiamati letti assegnati) si possano destinare ad una specialità chirurgica piuttosto che a un'altra. Ogni volta che si assegna questa specialità, si aggiorna il peso restante per quella specialità.

- Scelta la specialità, si scorre la lista d'attesa finché non si trova un paziente con quella specialità (quindi si prende quello con IdPrenotazione inferiore) e NON TIPO.
- Si controlla la durata dell'intervento del paziente scelto.
- Si selezionano gli altri pazienti sulla base del tempo restante e degli interventi che meglio si adattano sempre aggiornando lo slot libero rimanente.

Le decisioni e regole sopra descritte sono state prese in considerazione

Si procede allo stesso modo prima per le 4 sale occupabili da pazienti di priorità A e poi per i pazienti con diverse priorità, come definito in precedenza.

Questo programma è riadattabile anche in altri contesti, apportando modifiche di poco conto al codice.

Per poterlo utilizzare altrove, è necessario aggiornare:

- le specialità chirurgiche che operano in quel determinato ospedale in termini di tipologia (nome), numero e ore di sala operatoria assegnate;
- il numero delle sale operatorie;
- la lista operatoria cui fa riferimento quell'ospedale;
- il numero di posti in terapia intensiva;
- le festività di quella Regione o Comune;
- altre eventuali necessità emerse da quella determinata sede e/o eventuali modifiche o cancellazione di regole stabilite a monte nella scrittura del codice stesso.

Analisi programma in C++

L'interfaccia "Menù iniziale" chiede all'utente quante settimane voglia schedare e, sulla base dell'opzione scelta viene inserita la lista operatoria del giorno o della settimana corrente.

È previsto anche un comando che consente di passare al giorno seguente.

L'output viene generato in una cartella contenente tanti file quante sono le settimane richieste.

I giorni festivi, non prevedendo la programmazione operatoria in elezione, sono contrassegnati dalla data, dal giorno della settimana e dalla dicitura "Festivo".

Le restanti giornate feriali presentano la data, il giorno della settimana e la potenziale lista operatoria. Il potenziale programma operatorio è suddiviso per sala e, per ognuna di esse, è segnata la specialità chirurgica assegnata.

Il paziente è univocamente identificato dal nosologico (un numero progressivo), dalla priorità dell'intervento e dalla durata stimata dello stesso.

```
***** MENU INIZIALE *****
Data di oggi: 2024-01-01, Lunedì'

Digita uno dei seguenti numeri per selezionare una delle opzioni desiderate:
    1) Visualizzare la programmazione operatoria in data odierna
    2) Passa al giorno seguente
    3) Visualizzare la programmazione operatoria nella settimana corrente
    4) Exit
*****
Opzione scelta:
```

Figura 3: Menù iniziale del programma in C++

```
***** MENU INIZIALE *****
Data di oggi: 2024-01-01, Lunedì'

Digita uno dei seguenti numeri per selezionare una delle opzioni desiderate:
    1) Visualizzare la programmazione operatoria in data odierna
    2) Passa al giorno seguente
    3) Visualizzare la programmazione operatoria nella settimana corrente
    4) Exit
*****
Opzione scelta: 3

Programma settimanale calcolato. Puoi consultarlo nel file: nda_da_01-08_a_01-14.txt
Premi INVIO per continuare
```

Figura 4: Menù iniziale del programma in C++ con la didascalia successiva all'opzione selezionata

Giorno 2024-01-01 Lunedi' - B
FESTIVO!

Figura 5: Output generato dal programma in C++ in un giorno festivo

```
Giorno 2024-01-02 Martedi' - B
Sala 1 - (10) CH_VASC
paz: 0000000000000002, prio: A, durata: T(8) - pazTIPO
-----
numPaz: 1, slotLiberiRimanenti: 0

Sala 2 - (4) CH_OCU
paz: 0000000000000007, prio: A, durata: E(8) - pazTIPO
-----
numPaz: 1, slotLiberiRimanenti: 0

Sala 3 - (8) CH_PLA
paz: 0000000000000005, prio: A, durata: L(4)
paz: 0000000000000014, prio: A, durata: L(4)
-----
numPaz: 2, slotLiberiRimanenti: 0

Sala 4 - (3) CH_ODONT
paz: 0000000000000011, prio: A, durata: L(4)
paz: 0000000000000027, prio: A, durata: C(2)
paz: 00000000000000202, prio: A, durata: E(1)
paz: 00000000000000328, prio: A, durata: E(1)
-----
numPaz: 4, slotLiberiRimanenti: 0

Sala 5 - (9) CH_URO
paz: 0000000000000012, prio: A, durata: C(2)
paz: 0000000000000026, prio: A, durata: C(2)
paz: 0000000000000090, prio: A, durata: C(2)
paz: 00000000000000336, prio: A, durata: C(2)
-----
numPaz: 4, slotLiberiRimanenti: 0

Sala 6 - (14) TARF
paz: 0000000000000013, prio: A, durata: C(2)
paz: 0000000000000092, prio: A, durata: C(2)
paz: 00000000000000123, prio: A, durata: E(1)
paz: 00000000000000189, prio: A, durata: E(1)
paz: 00000000000000218, prio: A, durata: C(2)
-----
numPaz: 5, slotLiberiRimanenti: 0

Sala 7 - (12) ENDO_DIG
paz: 0000000000000001, prio: B, durata: L(4)
paz: 0000000000000044, prio: B, durata: E(1)
paz: 00000000000000125, prio: B, durata: E(1)
paz: 00000000000000130, prio: B, durata: C(2)
-----
numPaz: 4, slotLiberiRimanenti: 0

Sala 8 - (6) CH_ORTOP
paz: 0000000000000003, prio: B, durata: C(2)
paz: 0000000000000088, prio: B, durata: L(4)
paz: 00000000000000301, prio: B, durata: E(1)
paz: 00000000000000320, prio: B, durata: E(1)
-----
numPaz: 4, slotLiberiRimanenti: 0
```

Figura 6: Output giornaliero generato dal programma in C++

3.4. Un approccio di Programmazione Lineare e Ottimizzazione Combinatoria.

La maggior parte dei problemi di Ottimizzazione Combinatoria possono essere formulati mediante l'utilizzo di modelli di Programmazione Matematica e, generalmente, vengono tradotti in modelli di programmazione lineare intera MILP [23].

La Programmazione Matematica è una branca della Ricerca Operativa che, a sua volta, è una metodologia applicabile a svariati contesti che ha lo scopo di migliorare l'efficienza delle soluzioni da prendere.

La Programmazione Matematica prevede una funzione obiettivo che consenta di massimizzare o minimizzare una determinata quantità soggetta ad un insieme di vincoli stabiliti a monte che costituiscono i requisiti a cui la soluzione ricercata deve sottostare.

In Programmazione Lineare, obiettivo e vincoli devono essere tradotti in espressioni lineari delle variabili dove i vincoli sono rappresentati sotto forma di equazioni o disequazioni.

Un qualsiasi modello di programmazione lineare si articola nell'utilizzo di tre tipologie di variabili fondamentali [23][24][25]:

- Variabili decisionali: sono le incognite del problema, conoscerne il valore significa poter calcolare le soluzioni del problema e le relative funzioni di costo.
Le variabili possono essere reali e positive, intere e positive o binarie qualora possano assumere solo i valori 0 o 1.
- Vincoli: permettono di valutare l'ammissibilità di una soluzione trovata. Sono formulati mediante equazioni o disequazioni delle variabili. Talvolta può essere interessante rilassare uno o più vincoli e analizzare come varia la soluzione.
- Funzione obiettivo: è scritta in funzione delle variabili decisionali e può essere massimizzata o minimizzata in base a ciò che è necessario trovare. La funzione obiettivo va a calcolare il valore di tutte le soluzioni ammissibili, cioè compatibili con i vincoli e, dopo averle confrontate, permette di trovare quella ottima.

Una funzione di massimizzazione di un obiettivo può essere scritta come:

$$\max F(x_1, x_2, \dots, x_n) = c_1x_1, c_2x_2, \dots, c_nx_n$$

Una funzione di minimizzazione di un obiettivo può essere scritta come:

$$\min f(x_1, x_2, \dots, x_n) = c_1x_1, c_2x_2, \dots, c_nx_n$$

Dove le x_n rappresentano le variabili decisionali e c_n rappresentano i coefficienti di costo legato all'utilizzo di tale variabile [23][24].

3.5. Descrizione del modello di programmazione lineare del problema proposto

Per il problema in questione, obiettivo, variabili e vincoli si esprimono in programmazione lineare nel modo seguente [34].

Funzione obiettivo: massimizzare la somma pesata degli interventi eseguiti al fine di minimizzare gli slot operatori liberi schedulando gli interventi delle relative specialità.

$$\max \sum_i \sum_{sala} \sum_{slot} (peso_i \cdot durata_i \cdot x_{i,sala,slot})$$

Variabili:

- $x_{i,sala,slot}$ binaria: l'intervento i viene assegnato ad una sala in uno slot.
- $y_{specialita,sala,slot}$ binaria: nella sala operatoria opera una specialità s .

Gli interventi avranno pesi differenti in base alla priorità. Il peso è un coefficiente che è stato stimato con lo scopo di dare massima priorità agli interventi di pazienti che necessitano la Terapia Intensiva Post-Operatoria e a seguire dare una priorità superiore agli interventi di priorità A, a discapito di quelli di priorità B, C o D.

- $peso_{i \text{ priorit\`a } T} = 100$; possono avere solo durata TIPO.
- $peso_{i \text{ priorit\`a } A} = 10$; possono essere lunghi, corti o di anestesia locale.
- $peso_{i \text{ priorit\`a } B} = peso_{i \text{ priorit\`a } C} = peso_{i \text{ priorit\`a } D} = 1$; possono essere lunghi, corti o di anestesia locale.

La settimana lavorativa è di 5 giorni a settimana. La giornata lavorativa 8:00-20:00 è divisa negli slot mattina e pomeriggio dei singoli giorni: 8:00-14:00 e 14:00-20:00. Il turno della

mattina e quello del pomeriggio sono entrambi di 6 ore, 360 minuti l'uno.

La durata lorda degli interventi è un coefficiente ed è stata calcolata nel seguente modo:

$$\circ \text{durata}_i \text{ anestesia locale} = \frac{12 \frac{\text{ore}}{\text{giorno}}}{8 \frac{\text{interventi}}{\text{giorno}}} = 1,5 \frac{\text{ore}}{\text{intervento}} = 90 \frac{\text{minuti}}{\text{intervento}}$$

$$\circ \text{durata}_i \text{ corto} = \frac{12 \frac{\text{ore}}{\text{giorno}}}{4 \frac{\text{interventi}}{\text{giorno}}} = 3 \frac{\text{ore}}{\text{intervento}} = 180 \frac{\text{minuti}}{\text{intervento}}$$

$$\circ \text{durata}_i \text{ lungo} = \frac{12 \frac{\text{ore}}{\text{giorno}}}{2 \frac{\text{interventi}}{\text{giorno}}} = 6 \frac{\text{ore}}{\text{intervento}} = 360 \frac{\text{minuti}}{\text{intervento}}$$

$$\circ \text{durata}_i \text{ TIPO} = \frac{12 \text{ ore/giorno}}{1 \text{ interventi/giorno}} = 12 \frac{\text{ore}}{\text{intervento}} = 720 \frac{\text{minuti}}{\text{intervento}}$$

- In un determinato slot possono svolgersi al più 4 interventi (in anestesia locale).

$$\sum_{\text{sala}} \sum_{\text{slot}} x_{i,\text{sala},\text{slot}} \leq 4 \quad \forall_i$$

- La somma delle durate degli interventi su una sala può ammontare al più a 720 minuti (giornata di lavoro da 12 ore).

$$\sum_i \sum_{\text{sala}} x_{i,\text{sala},\text{slot}} \cdot \text{durata}_i \leq 720 \quad \forall_{\text{sala}}$$

- Ogni sala può ospitare una ed una sola specialità.

$$\sum_{\text{specialita}} y_{\text{specialita},\text{sala},\text{slot}} = 1 \quad \forall_{\text{sala}}$$

- Può essere operato un paziente di una determinata specialità solo se quella stessa specialità è assegnata a quello slot operatorio al fine di garantire l'attrezzatura corretta in una determinata sala in un dato slot.

$$x_{i,\text{sala},\text{slot}} \leq y_{\text{specialita},\text{sala},\text{slot}}$$

- Su ogni sala opera una specialità differente.

$$y_{specialita1,sala,slot} + y_{specialita2,sala,slot} \leq 1 \quad \forall_{sala, specialita1 \neq specialita2}$$

- Possono essere operati massimo due pazienti di tipo T al giorno.

$$\sum_{i \text{ di tipo "T"}} \sum_{sala} x_{i,sala,slot} \leq 2 \quad \forall_{sala}$$

- Possono essere operati massimo 24 pazienti di tipo A al giorno. Vengono dedicate 4 sale alla priorità A per cui il limite dovrebbe essere 16 pazienti in ogni slot ma, poiché ci potrebbero essere fino a due sale TIPO libere per mancanza di pazienti TIPO, si aggiungono 8 pazienti al giorno.

$$\sum_{i \text{ di tipo "A"}} \sum_{sala} \sum_{slot} x_{i,sala,slot} \leq 24 \quad \forall_{sala}$$

- Possono essere operati massimo 32 pazienti sommando gli interventi di priorità B, C o D. Vengono dedicate 2 sale a queste priorità, per cui il limite dovrebbe essere 8 ma, poiché ci potrebbero essere fino a due sale TIPO libere per mancanza di pazienti TIPO, e fino a 4 sale dedicate alla priorità A, si aggiunge la restante quota parte.

$$\begin{aligned} & \sum_{i \text{ di tipo "B"}} \sum_{sala} \sum_{slot} x_{i,sala,slot} + \sum_{i \text{ di tipo "C"}} \sum_{sala} \sum_{slot} x_{i,sala,slot} \\ & + \sum_{i \text{ di tipo "D"}} \sum_{sala} \sum_{slot} x_{i,sala,slot} \leq 32 \quad \forall_{sala} \end{aligned}$$

- Il numero di sale utilizzabili è inferiore o uguale a otto.

$$\sum_{sala} y_{specialita,sala,slot} \leq 8 \quad \forall_{sala}$$

- Il numero di slot operatori per sala è pari a due al giorno: mattina e pomeriggio.

$$\sum_i \sum_{slot} x_{i,sala,slot} \leq 2 \quad \forall_{sala}$$

3.6. Descrizione del Modello Mosel

Mosel è un linguaggio di programmazione e di modellazione ad alto livello che consente una facile esecuzione di algoritmi di programmazione matematica.

Viene utilizzato in applicazioni pratiche, al fine di risolvere problemi di ottimizzazione che vanno dal settore bancario, all'e-commerce, alla supply chain e produzione, grazie alla versatilità e alla notazione molto vicina alla formulazione scritta con simboli matematici su carta.

Il modello di programmazione lineare è stato eseguito sul software FICO@Xpress Optimization [29][30] che consente di risolvere problemi ad alta complessità e, dunque, con numerose variabili in pochi minuti.

Nella ricerca di soluzioni ammissibili, ma soprattutto dell'ottimo della funzione, il tempo è un fattore determinante.

In presenza di problemi di ottimizzazione complessi, con quantità considerevoli di variabili e di vincoli, i programmi tendono infatti a non riuscire ad elaborare una soluzione real time. Xpress utilizza un approccio branch&bound per cercare lo spazio di rilassamento dei vincoli per soluzioni Mixed Integer Programming (MIP).

Il branch&bound è una tecnica generale per la risoluzione di problemi con spazio di soluzioni finito: scompone il problema originale in sottoproblemi, valuta ed enumera le soluzioni e dà luogo ad un albero delle soluzioni ammissibili [23].

Vengono generati, dunque, i vari sottoproblemi se ne verifica l'ammissibilità e si valuta l'insieme delle soluzioni ammissibili, al fine di scegliere quella corrispondente al valore ottimo.

Il processo di ramificazione del problema iniziale in sottoproblemi è detto branching.

Un nodo principale si ramifica in altri sottoproblemi, rappresentati da nodi, e collegati al problema principale da archi, rappresentati da frecce.

Il risultato ottenuto è l'albero decisionale, anche chiamato branch decision tree, rappresentato nella figura sottostante [23][24].

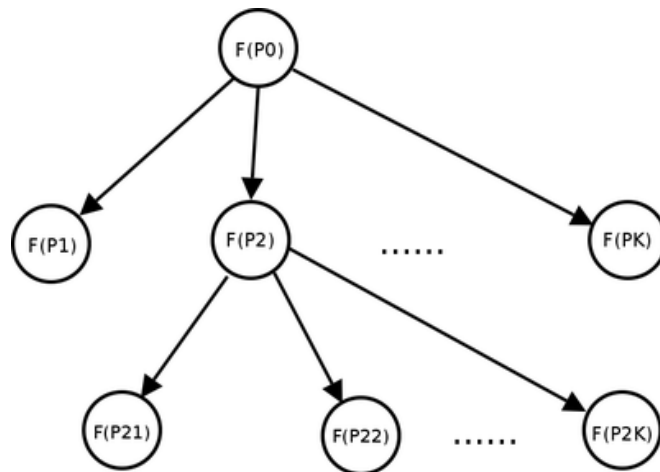


Figura 7: Branch decision tree

Successivamente si valuta la qualità di tutti i sottoproblemi generati, calcolandone il valore limite per la funzione obiettivo: operazione di bound.

In alcuni casi, è possibile che si generi un numero elevatissimo di problemi, il che rende complicato l'utilizzo di questo metodo ricorsivo.

Al metodo branch&bound, infatti, viene associato il “Criterio di Stop”, che consiste nella ricerca della miglior soluzione ammissibile trovata in tempi definiti: se non venissero chiusi o esplorati tutti i nodi entro questi limiti di tempo la soluzione trovata non sarebbe necessariamente l'ottimo [23].

A questo proposito su Mosel è stato introdotto il comando `setparam("XPRS_MAXTIME",120)`, valutando le diverse soluzioni trovate in differenti intervalli di tempo.

Il problema gestisce oltre decine di migliaia di variabili e vincoli per cui, inevitabilmente, più grande è l'intervallo di tempo scelto, più è probabile che la soluzione tenda all'ottimo globale.

Osservando l'output ottenuto dal solver, si possono notare due valori distinti relativi al Best Bound e Best Solution.

La differenza tra “Best Bound” e “Best Solution” è significativa nell'ambito della risoluzione dei problemi di ottimizzazione lineare.

Per Best Bound si intende il limite superiore (nei problemi di massimo), ossia il valore ottimo del rilassamento continuo del problema, ossia del modello di programmazione lineare in questione cui siano stati tolti (rilassati) i vincoli di interezza delle variabili. Corrispondentemente, il Best Bound non rappresenta necessariamente una soluzione ammissibile [23].

La Best Solution rappresenta, invece, il valore ottenuto dall'algoritmo nell'intervallo richiesto e soddisfa tutti i vincoli del problema e, dunque, è la migliore tra le soluzioni ammissibili trovate.

Idealmente, al fine di poter affermare di essere in presenza di un ottimo, Best Solution e Best Bound devono avere uno scarto minimo.

DATI INPUT MOSEL

Per poter svolgere le analisi senza sovraccaricare il sistema, il campione di osservazione è stato ristretto a 500 pazienti.

I pazienti, univocamente contraddistinti dal nosologico progressivo, sono stati selezionati dalla lista d'attesa.

Per ciascuno di essi sono stati riportati: la specialità chirurgica di appartenenza, la durata prevista dell'intervento e il peso del relativo intervento.

Il file txt da cui vengono presi i dati di input è strutturato come segue:

- La prima riga contiene due numeri: il campione selezionato pari a 500 pazienti, e la capacità massima di uno slot espressa in minuti.
La sala è utilizzabile per due slot da 360 minuti, per un totale di due slot per sala.
- La seconda riga contiene la specialità chirurgica di appartenenza del paziente, numerata da 1 a 16.
- La terza riga contiene la durata stimata dell'intervento sulla base dello storico degli interventi. Questi si suddividono in base alla durata, pari a 90, 180, 360 o 720 minuti.
- La quarta e ultima riga rappresenta la codifica della priorità, identificata dal peso dell'intervento nella funzione obiettivo, come descritto precedentemente.

Considerazioni

Entrambi i modelli forniscono una soluzione relativa alla schedulazione delle sale operatorie. La programmazione operatoria fornita dall'output, dei due modelli, deve essere intesa come uno strumento di supporto per il Coordinatore Infermieristico, il quale può effettuare delle modifiche, sia in termine di ordine, sia in termine di sostituzione di paziente per necessità di vario genere.

Questi strumenti di supporto, sulla base delle esigenze previste dal programma operatorio, aiutano il comparto a distribuire e gestire con maggiore consapevolezza le ferie e i permessi degli infermieri, compresi gli eventuali scambi di turno tra specialità.

Aiutano anche a tracciare con facilità la tipologia di intervento, le caratteristiche del paziente e la capacità occupata della sala operatoria.

A questo proposito, è stata inserita anche la voce "Slot liberi" che sta ad indicare che la sala non è stata occupata per un certo intervallo di tempo, in quanto non era presente alcun paziente nella lista che soddisfacesse i criteri imposti dal codice.

In questi casi, l'intervento migliorativo del Coordinatore degli infermieri svolge un ruolo fondamentale, consentendo la possibilità di effettuare degli "incastri" non previsti dal programma, che generino ulteriore ottimizzazione.

4. Risultati

4.1. Analisi e confronto dei risultati ottenuti

Lo scopo di questo elaborato consiste nello sviluppo, nel confronto e nella comprensione dei risultati ottenuti dai due approcci di soluzione (l'algoritmo costruttivo e l'approccio basato su programmazione lineare), che hanno come obiettivo finale quello di supportare la pianificazione degli interventi, offrendo una loro allocazione ottimizzata in tempi limitati.

I programmi utilizzati, seppur perseguano lo stesso scopo, arrivano alla programmazione richiesta in metodi differenti.

Il problema preso in esame è un problema di ottimizzazione combinatoria.

Il solver di ottimizzazione impiegato per valutare la soluzione del codice di programmazione lineare è stato Xpress IVE.

Sulla base di vincoli prefissati, l'algoritmo scritto su C++ risolve il problema in maniera costruttiva.

I pazienti vengono assegnati alla sala in maniera progressiva fino al raggiungimento massimo della capacità della sala stessa, 720 minuti, laddove possibile.

Questo modus operandi non genera necessariamente una soluzione ottima.

Il solver utilizzato per ottenere l'output di questo programma è stato CodeBlocks.

Lo stesso problema viene affrontato dunque in due modi completamente diversi ed è dunque possibile verificare la consistenza e le differenze tra le due soluzioni generate.

Questi modelli sono di supporto alla decisione ma non sono pensati per sostituirsi al personale.

Il contesto ospedaliero è un contesto altamente variabile e una determinata operazione pianificata potrebbe non venire eseguita per molteplici ragioni: problemi logistici o di salute legati al paziente, emergenze ecc.

Gli approcci di risoluzione proposti, dunque, generano output che rappresentano una possibile decisione, ma non si sostituiscono al decisore.

4.1.1. Output grafici ottenuti dall'algoritmo costruttivo in C++

Per confrontare i risultati ottenuti dall'algoritmo costruttivo, con i risultati ottenuti dalla programmazione lineare, è stato necessario rielaborare gli output delle quattro settimane.

Di ciascun giorno, visualizzato come in *Figura 6*, sono state selezionate le informazioni riguardanti a ogni intervento svolto in ciascuna sala (numero e priorità chirurgica).

Successivamente, sono state create due tabelle:

- La prima rappresentativa del numero di interventi svolto nelle singole sale nei differenti lassi di tempo considerati;
- La seconda rappresentativa del peso di quei determinati interventi per la sua durata, in quella data sala e in quel dato giorno.

Sono state assegnate, a rotazione giorno per giorno, due sale ai pazienti Tipo, quattro alle operazioni di priorità A e due sale agli interventi di priorità B, C e D.

I pesi considerati sono stati definiti come segue:

- a ogni intervento in TIPO è stato assegnato un valore pari a 100;
- a ogni intervento di priorità A è stato assegnato un peso di 10 unità;
- a ogni paziente con priorità B, C e D, un peso pari a 1.

Questi pesi sono stati pensati per dare maggiore rilevanza ad interventi ad alta priorità, poiché devono essere svolti in tempi più stringenti in quanto considerati urgenze, seppur differibili.

Infine, i risultati ottenuti dalla funzione obiettivo non sono altro che la sommatoria pesata dei singoli interventi svolti sulle otto sale, moltiplicati per la durata e per il peso dell'intervento.

La scelta di utilizzare dei coefficienti indotti dalla moltiplicazione tra durata e peso è legata alla volontà di non voler dare maggiore priorità ad interventi corti rispetto a quelli lunghi.

Nella tabella sottostante [*Figura 10*] rappresenta la soluzione dell'algoritmo costruttivo in termini di numero di interventi schedulati per sala e per giorno in un intervallo di tempo considerato pari a 4 settimane.

NUMERO PAZIENTI OPERATI								
	SALA 1 Tipo	SALA 2 - Tipo	SALA 3 - A	SALA 4 - A	SALA 5 - A	SALA 6 - A	SALA 7 - B/C/D	SALA 8 - B/C/D
GIORNO 1	1	1	2	4	4	5	4	4
GIORNO 2	1	1	3	3	4	2	5	5
GIORNO 3	1	1	4	3	3	4	4	4
GIORNO 4	1	1	2	3	2	3	5	5
GIORNO 5	1	1	3	2	3	2	4	2
GIORNO 6	1	1	3	4	2	3	4	2
GIORNO 7	1	1	2	2	4	2	4	6
GIORNO 8	1	1	4	3	3	4	4	2
GIORNO 9	1	1	4	3	4	3	3	3
GIORNO 10	1	1	2	3	5	4	2	3
GIORNO 11	1	1	2	3	2	3	1	2
GIORNO 12	1	1	2	2	6	5	4	2
GIORNO 13	1	1	5	3	2	3	1	5
GIORNO 14	1	1	4	4	2	2	3	4
GIORNO 15	1	1	2	4	2	2	4	1
GIORNO 16	1	1	4	2	4	4	2	4
GIORNO 17	1	1	2	3	2	5	4	3
GIORNO 18	1	1	2	3	3	2	3	2
GIORNO 19	1	1	2	4	5	4	2	1
GIORNO 20	1	1	3	1	3	2	5	4

Figura 10: Numero interventi considerati nei 20 giorni

Moltiplicando il numero di interventi nelle diverse sale per i pesi assegnati, sulla base della priorità e della durata, è stata ottenuta la tabella seguente [Figura 11]:

PESI TOTALI								
	SALA 1 Tipo	SALA 2 - Tipo	SALA 3 - A	SALA 4 - A	SALA 5 - A	SALA 6 - A	SALA 7 - B/C/D	SALA 8 - B/C/D
GIORNO 1	72000	72000	7200	7200	7200	6390	720	720
GIORNO 2	72000	72000	7200	5580	2340	7200	720	720
GIORNO 3	72000	72000	6390	7200	5580	7200	720	720
GIORNO 4	72000	72000	7200	5580	7200	7200	720	720
GIORNO 5	72000	72000	7200	7200	7200	7200	720	720
GIORNO 6	72000	72000	7200	5490	7200	7200	720	720
GIORNO 7	72000	72000	7200	7200	7200	7200	720	720
GIORNO 8	72000	72000	7200	7200	7200	3960	720	720
GIORNO 9	72000	72000	2340	7200	7200	6300	720	720
GIORNO 10	72000	72000	7200	7200	7200	7200	720	720
GIORNO 11	72000	72000	7200	7200	7200	5580	360	720
GIORNO 12	72000	72000	7200	7200	6390	4770	720	720
GIORNO 13	72000	72000	7200	5580	7200	5580	180	630
GIORNO 14	72000	72000	7200	7200	7200	7200	630	720
GIORNO 15	72000	72000	7200	5580	7200	7200	720	180
GIORNO 16	72000	72000	7200	7200	2160	5580	720	720
GIORNO 17	72000	72000	7200	7200	7200	5580	720	720
GIORNO 18	72000	72000	7200	7200	7200	7200	450	360
GIORNO 19	72000	72000	7200	7200	7200	7200	720	360
GIORNO 20	72000	72000	7200	3600	7200	3960	720	720

Figura 11: Pesi totali nelle diverse sale nei 20 giorni

4.1.2. Output grafici generati da Mosel

Le immagini sottostanti sono inerenti ad una delle 30 istanze discusse successivamente. In particolare, riguardano la casistica di 500 operazioni con 16 specialità considerate.

Ciascuna immagine rappresenta uno dei lassi di tempo considerato, rispettivamente una, due, tre o quattro settimane lavorative.

Si può osservare come in tutti e 4 gli specchietti si registra l'ottimo, raggiunto in tempi differenti.

Stats		Stats	
Matrix:		Presolved:	
Rows(constraints): 40980		Rows(constraints): 26283	
Columns(variables): 41280		Columns(variables): 38908	
Nonzero elements: 205200		Nonzero elements: 160972	
Global entities: 41280		Global entities: 38908	
Sets: 0		Sets: 0	
Set members: 0		Set members: 0	
Overall status: Finished global search.			
LP relaxation:		Global search:	
Algorithm: Barrier		Current node: 1	
Simplex iterations: 883		Depth: 1	
Objective: 288000		Active nodes: 0	
Status: Unfinished		Best bound: 208000	
Time: 3.0s		Best solution: 288000	
		Gap: 0%	
		Status: Solution is optimal.	
		Time: 3.7s	
Time overheads:			
Progress graphs: 0.6s			
Writing output: 0.1s			
Pausing: 0.0s			
Updating status: 0.6s			
Output/Input Stats Matrix Solutions Objective MIP search BB tree User graph IIS			

Stats		Stats	
Matrix:		Presolved:	
Rows(constraints): 81460		Rows(constraints): 52115	
Columns(variables): 82560		Columns(variables): 77816	
Nonzero elements: 410400		Nonzero elements: 321944	
Global entities: 82560		Global entities: 77816	
Sets: 0		Sets: 0	
Set members: 0		Set members: 0	
Overall status: Finished global search.			
LP relaxation:		Global search:	
Algorithm: Simplex primal		Current node: 1	
Simplex iterations: 15200		Depth: 1	
Objective: 576000		Active nodes: 0	
Status: Unfinished		Best bound: 576000	
Time: 23.1s		Best solution: 576000	
		Gap: 0%	
		Status: Solution is optimal.	
		Time: 37.6s	
Time overheads:			
Progress graphs: 2.4s			
Writing output: 0.1s			
Pausing: 0.0s			
Updating status: 2.4s			
Output/Input Stats Matrix Solutions Objective MIP search BB tree User graph IIS			

Figura 8: Output Mosel: 500 interventi, 16 specialità chirurgiche, 5 e 10 giorni

Stats		Stats	
Matrix:		Presolved:	
Rows(constraints): 121940		Rows(constraints): 77947	
Columns(variables): 123840		Columns(variables): 116724	
Nonzero elements: 615600		Nonzero elements: 482916	
Global entities: 123840		Global entities: 116724	
Sets: 0		Sets: 0	
Set members: 0		Set members: 0	
Overall status: Finished global search.			
LP relaxation:		Global search:	
Algorithm: Simplex primal		Current node: 1	
Simplex iterations: 0		Depth: 1	
Objective: 637200		Active nodes: 0	
Status: Unfinished		Best bound: 637200	
Time: 15.7s		Best solution: 637200	
		Gap: 0%	
		Status: Solution is optimal.	
		Time: 44.9s	
Time overheads:			
Progress graphs: 3.7s			
Writing output: 0.1s			
Pausing: 0.0s			
Updating status: 3.7s			
Output/Input Stats Matrix Solutions Objective MIP search BB tree User graph IIS			

Stats		Stats	
Matrix:		Presolved:	
Rows(constraints): 162420		Rows(constraints): 103779	
Columns(variables): 165120		Columns(variables): 155632	
Nonzero elements: 820800		Nonzero elements: 643888	
Global entities: 165120		Global entities: 155632	
Sets: 0		Sets: 0	
Set members: 0		Set members: 0	
Overall status: Finished global search.			
LP relaxation:		Global search:	
Algorithm: Barrier		Current node: 1	
Simplex iterations: 7344		Depth: 1	
Objective: 650430		Active nodes: 0	
Status: Unfinished		Best bound: 650430	
Time: 18.0s		Best solution: 650430	
		Gap: 0%	
		Status: Solution is optimal.	
		Time: 74.5s	
Time overheads:			
Progress graphs: 3.0s			
Writing output: 0.1s			
Pausing: 0.0s			
Updating status: 3.0s			
Output/Input Stats Matrix Solutions Objective MIP search BB tree User graph IIS			

Figura 9: Output Mosel: 500 interventi, 16 specialità chirurgiche, 15 e 20 giorni

4.1.3. Istanze generate

Sono state generate 28 istanze differenti inducendo tre tipologie di analisi su livelli diversi.

Le prime due tipologie di analisi mostrano il confronto dei valori ottenuti dalle funzioni obiettivo valutando gli output ottenuti da Mosel e dall'algoritmo costruttivo utilizzando due liste d'attesa distinte fornite dall'Ospedale, entrambe contenenti 500 pazienti.

Sono state generate righe differenti per confrontare gli output ottenuti in lassi di tempo differenti pari a 5, 10, 15 e 20 giorni.

Il programma in Mosel, nonostante le 156 mila variabili e oltre 106 mila vincoli in gioco, è riuscito a generare soluzioni di ottimo in tempi accettabili, variabili da pochi secondi fino a meno di 4 minuti.

All'aumentare del tempo considerato, il gap tra Best Bound e Best Solution diminuisce fino a trovare l'ottimo, istante nel quale si annulla.

Data la prontezza di risposta del solver, quest'ultimo non è stato vincolato a dare una soluzione in tempi prefissati.

L'algoritmo costruttivo in C++ ha fornito i risultati in tempi trascurabili. Il numero degli interventi, giorno per giorno, è stato inserito in un foglio Excel ed è stato moltiplicato per il relativo peso legato alla priorità e per la durata dell'intervento stesso.

La prima e la seconda analisi differiscono dal numero di specialità chirurgiche coinvolte.

Nel primo caso sono state valutate due liste d'attesa differenti ma sono state considerate le stesse specialità chirurgiche.

La percentuale di scostamento, e conseguentemente la percentuale di errore che si ottiene dall'utilizzo di Mosel rispetto all'utilizzo dell'algoritmo costruttivo in C++, è decrescente al crescere dei giorni considerati. Questo è prevedibile considerando una lista d'attesa pari a 500 pazienti. Considerando che in una giornata vengono operati circa 20/25 degenti, tra operazioni lunghe e operazioni in anestesia locale, in una ventina di giorni la lista d'attesa viene quasi portata a termine e, dunque, i due valori tendono ad avvicinarsi.

La seconda lista d'attesa dimostra come le considerazioni fatte a monte siano robuste. Le percentuali d'errore in entrambi i casi sono pressoché le stesse.

RISULTATI DELL'ANALISI								
	Giorni considerati	N. specialità chirurgiche	N. pazienti in lista	Obiettivo in Mosel	Tempi in Mosel	Obiettivo in C++	Tempi in C++	Errore%
Lista attesa 1	5	16	500	288000	3,7 s	214965	trascurabili	25%
	10	16	500	576000	37,6 s	430087	trascurabili	25%
	15	16	500	637200	44,9 s	516042	trascurabili	19%
	20	16	500	650430	74,5 s	572550	trascurabili	12%
Lista attesa 2	5	16	500	288000	3,8 s	220965	trascurabili	23%
	10	16	500	576000	22,0 s	459087	trascurabili	20%
	15	16	500	659070	62,3 s	576002	trascurabili	13%
	20	16	500	671130	203,0 s	589750	trascurabili	13%

Figura 12: Analisi 1 – confronto risultati funzione obiettivo

La seconda analisi confronta due liste d'attesa differenti: prendendo la prima lista d'attesa, sono stati sostituiti gli interventi di terapia antalgica, con altri interventi di altre specialità chirurgiche.

Il numero di specialità considerato nel primo blocco è pari a 15 specialità, nel secondo ha 16.

In Mosel la funzione obiettivo ottenuta risulta essere la stessa per ambo i casi; questo è giustificabile dal fatto che gli interventi della terapia antalgica occupano circa il 4% della lista completa.

La funzione obiettivo ottenuta dai risultati del C++ è differente ma gli scostamenti sono simili a quelli della prima analisi.

RISULTATI DELL'ANALISI								
	Giorni considerati	N. specialità chirurgiche	N. pazienti in lista	Obiettivo in Mosel	Tempi in Mosel	Obiettivo in C++	Tempi in C++	Errore%
Lista attesa 3 - 15 specialità	5	15	500	288000	3,8 s	213765	trascurabili	25%
	10	15	500	576000	22,0 s	429989	trascurabili	25%
	15	15	500	637200	62,3 s	516042	trascurabili	19%
	20	15	500	650430	203,0 s	571320	trascurabili	12%
Lista attesa 1 - 16 specialità	5	16	500	288000	3,7 s	214965	trascurabili	25%
	10	16	500	576000	37,6 s	430087	trascurabili	25%
	15	16	500	637200	44,9 s	516042	trascurabili	19%
	20	16	500	650430	74,5 s	572550	trascurabili	12%

Figura 13: Analisi 2 – confronto risultati funzione obiettivo

La terza analisi differisce dalle precedenti due in quanto elabora solamente i dati ottenuti da Mosel in caso di differenti liste d'attesa.

Il numero di pazienti osservati è differente mentre rimane costante il numero di specialità chirurgiche considerate.

Il confronto viene svolto osservando solamente il risultato delle diverse funzioni obiettivo

nei vari intervalli di tempo che vanno da una fino a quattro settimane.

- La lista d’attesa contenente 420 pazienti gestisce un numero di vincoli pari a 130 mila variabili e 85,5 mila vincoli;
- La lista d’attesa contenente 500 operazioni gestisce quasi 155,6 mila variabili e 104 mila vincoli;
- La lista d’attesa da 600 pazienti gestisce oltre 186 mila variabili e 121 mila vincoli.

È osservabile come all’aumentare delle variabili aumenta il tempo di risposta del solver, seppure questo ottenga il risultato finale in tempi che vanno poco oltre un paio di minuti.

RISULTATI DELL'ANALISI					
	Giorni considerati	N. specialità chirurgiche	N. pazienti in lista	Obiettivo in Mosel	Tempi in Mosel
Lista attesa 420 pazienti	5	16	420	288000	3,1s
	10	16	420	509580	17,4
	15	16	420	536940	50,9 s
	20	16	420	536940	22,1 s
Lista attesa 500 pazienti	5	16	500	288000	3,7 s
	10	16	500	576000	37,6 s
	15	16	500	637200	44,9 s
	20	16	500	650430	74,5 s
Lista attesa 600 pazienti	5	16	600	288000	5,6 s
	10	16	600	576000	21,7 s
	15	16	600	737640	69,3 s
	20	16	600	766440	139,7 s

Figura 14: Analisi 3 - confronto risultati funzione obiettivo

4.2. Limiti delle soluzioni proposte

Gli algoritmi offerti sono stati pensati su misura per l'esperienza relativa all'ospedale di Sassari.

Ogni realtà ha infatti le proprie regole e le proprie necessità: ciò che può essere ottimale per un ospedale, può non esserlo per un altro e, dunque, potrebbe essere necessario cambiare alcuni vincoli e parametri prestabiliti.

Queste soluzioni possono essere utilizzate seguendo una logica di tipo what-if-analysis al variare dei vincoli e dei parametri in gioco.

Lo strumento di "what-if analysis" è in grado di elaborare scenari differenti, per offrire i diversi esiti possibili.

A differenza delle analisi predittive evolute, richiede pochi dati di base per poter essere elaborata.

Si possono modificare o rilassare alcuni vincoli, al fine di valutare i cambiamenti dell'output.

A questo proposito, l'informatizzazione dei dati diventa, invece, fondamentale, per consentire trasparenza, tracciabilità e possibili analisi su scenari differenti.

Conclusioni

Questo lavoro è il frutto dello studio e della ricerca nell'ambito della gestione delle sale operatorie e, in particolare, dell'ottimizzazione dei posti in terapia intensiva post-operatoria nell'Ospedale Stecca Bianca di Sassari.

L'analisi è stata svolta in tre fasi: nella prima fase è stato studiato da vicino il contesto e ne sono state individuate le criticità; successivamente, nella seconda fase, sono stati presi in considerazione i vincoli della struttura, le necessità del comparto e dei chirurghi; infine sono stati sviluppati e messi a confronto due algoritmi per la schedulazione degli interventi chirurgici.

I modelli teorici sono stati testati con istanze reali al fine di valutare la qualità delle soluzioni ottenute.

Gli algoritmi hanno utilizzato approcci differenti, uno costruttivo e uno puntuale.

Il primo, scritto in C++, ha ricercato e trovato soluzioni real time anche se si discostano dall'ottimo.

L'algoritmo in Mosel, invece, ha ricercato e trovato l'ottimo in tempi ragionevoli anche in presenza di moli di dati rilevanti.

I risultati ottenuti hanno consentito di affermare che ci siano consistenti margini di miglioramento con l'utilizzo di entrambi i solver in termini di abbattimento delle liste d'attesa.

L'ottimizzazione delle risorse non deve in alcun modo intaccare il ruolo delle figure sanitarie, parte integrante dell'ingranaggio, ma può essere di supporto e alleggerire il carico di lavoro di alcune figure professionali.

Appendice

APPENDICE A

Nella seguente sezione viene proposto il modello in linguaggio C++ per intero, comprensivo di ogni passaggio elencato nel Capitolo precedente.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <list>
#include <cstdint>
#include <iomanip>
#include <limits>

using namespace std;

// Dichiarazioni di costanti
#define NUM_GIORNI_ANNO 366
#define NUM_GIORNI_SETT 7
#define NUM_SALE_TUTTE 8
#define NUM_SALE_PAZ_TIPO 2
#define NUM_SALE_PAZ_ELEZ_NON_A 2
#define NUM_SPECIALITA 16
#define SLOT_GIORNALIERI_PER_SALA 8
#define SLOT_OPER_TIPO 8
#define SLOT_OPER_LUNGA 4
#define SLOT_OPER_CORTA 2
#define SLOT_OPER_LOCALE 1
```


// Dichiarazioni di tipi enumerati

enum Specialita{

NESSUNA = -1,

BU = 0,

CH_GEN = 1,

CH_MXF = 2,

CH_ODONT = 3,

CH_OCU = 4,

CH_ORL = 5,

CH_ORTOP = 6,

CH_PED = 7,

CH_PLA = 8,

CH_URO = 9,

CH_VASC = 10,

EBUS = 11,

ENDO_DIG = 12,

PI_DIAB = 13,

TARF = 14,

T_ANT = 15

};

enum Priorita{ A=4, B=3, C=2, D=1};

// Dichiarazioni di strutture

struct Paziente{

unsigned long long int id; //può contenere numeri nell'intervallo [0, 18.446.744.073.709.551.615] (numeri fino a 20 cifre)

Specialita specialita;

Priorita priorita;

char durata_operazione;

};

```
struct PazienteTIPO{
    unsigned long long int id;
    Specialita specialita;
    Priorita priorita;
    char durata_operazione;
};
```

```
struct Giorno{
    string data;
    uint8_t ggsett;
    bool ggfestivo;
};
```

```
struct DatiOperazione{
    bool operata;
    short int num_giorno;
};
```

```
struct Sala{
    Specialita specialitaAssegnata = NESSUNA;
    uint8_t slotLiberiRimanenti = SLOT_GIORNALIERI_PER_SALA;
    uint8_t pazOperati = 0;
};
```

```
struct Letti{
    Specialita nome;
    uint8_t totLettiDisponibili;
    uint8_t numLettiRimanenti;
    double percLettiRimanenti;
};
```

```
//----- VARIABILI GLOBALI -----
```

```

//solo inizializzazione - una tantum
list<Paziente> pazienti;
list<PazienteTIPO> pazientiTIPO;
Giorno calendario[NUM_GIORNI_ANNO];
short int giornoCorrente = -1;
Priorita prioritaNONACercataOggi = D;
bool SpecialitaOperateRecentemente[NUM_SPECIALITA];

//reset ogni settimana
Letti agendaLetti[NUM_SPECIALITA];
string nomeFileOutput;

//reset ogni giorno
bool specialitaOperateOggi[NUM_SPECIALITA];
Sala agendaSaleGiornaliera[NUM_SALE_TUTTE];
int numTOTSaleProgrammateOggi = 0;

//————— DICHIARAZIONE FUNZIONI —————
//-----FUNZIONI DI INIZIALIZZAZIONE (dichiarazione) -----
-----
void inicializzaAgendaSaleGiornaliera();
int inicializzaAgendaLetti();
int costruisciAgenda2024();
int leggiListaAttesa();

//-----FUNZIONI PRINCIPALI (dichiarazione) -----
-----
void nuovoGiorno();
int programmaTUTTEOperazioniGiornaliere();

//pazienti TIPO
int programmaOperazioniPazTIPO(int);

```

```

//pazienti Elezione A
int programmaOperazioniPazElezione_A(int);

//pazienti Elezione NON A
Priorita prossimaPrioritaDaCercare_NON_A(Priorita);
int programmaOperazioniPazElezione_NON_A(int);

//in comune tra i pazienti Elezione A e NON A
Specialita scegliSpecialitaDaOperarePazElezione(Priorita prioritataCercata);

///-----FUNZIONI AUSILIARIE (dichiarazione) -----
list<Paziente> costruisciListaPazElezione_SpecEPrio(Specialita specDaOperare, Priorita
prioritaCercata);
void assegnaSalaASpecialita(Specialita);
void statisticheFineProgrammazioneSala(Specialita);
int cercaNumSala_Spec(Specialita);
bool checkPresenzaPazElezione_SpecEPrio(Specialita, Priorita);
unsigned int posizionePrimoPazElezione_SpecEPrio(Specialita, Priorita);
void eliminaPazElezioneDaListaAttesa_id(unsigned long long int);
void segnaOperazioneTIPO(Specialita, const PazienteTIPO&);
void segnaOperazioneElezione(Specialita, const Paziente&);
bool checkSpecialitaTIPO(Specialita);
bool checkSpecialitaElezione(Specialita);
void resetSpecialitaOperateRecentemente();
void resetSpecialitaOperateOggi();
void showMenu();
void scriviSuFileOutput(const string &str, const char&);
void cancellaFileOutput();
unsigned int riceviNumOperazione();
void premiInvioPerContinuare();

///-----FUNZIONI DI UTILITÀ (dichiarazione) -----
Specialita stringToSpecialita(const string&);

```

```

string specialitaToString(const Specialita&);
Priorita charToPriorita(const char&);
char prioritaToChar(const Priorita&);
string ggSettToString(const uint8_t&);
bool comparePazientiTIPO_id(const PazienteTIPO&, const PazienteTIPO&);
bool comparePazienti_id(const Paziente&, const Paziente&);
bool comparePazienti_priorita(const Paziente&, const Paziente&);
uint8_t durataInterventoPaziente(const Paziente&);
short int giornoPrecedente();
short int giornoSuccessivo(const short int&);
string estraiData(const string& dataCompleta);

```

```

//-----MAIN-----

```

```

int main() {
    unsigned int scelta;
    bool giaCalcolato = false;

    //inizializzazioni una tantum
    costruisciAgenda2024();
    leggiListaAttesa();
    resetSpecialitaOperateRecentemente();

    //inizializzazioni che faccio all'inizio e che poi rifarò altre volte
    resetSpecialitaOperateOggi();
    inizializzaAgendaSaleGiornaliera();
    inizializzaAgendaLetti();

    nuovoGiorno();

    while (true) {
        system("cls");
        showMenu();
    }
}

```

```

scelta = riceviNumOperazione();
cout<<endl;

switch (scelta) {
    case 1:
        //calcolo programma giornaliero
        if(!giaCalcolato) {
            programmaTUTTEOperazioniGiornaliere();
            cout<<"Programma giornaliero calcolato. "
            <<"Puoi consultarlo nel file: "<<nomeFileOutput.substr(10)<<endl<<endl;
            giaCalcolato = true;
        }
        else{
            cout<<"Programma gia' calcolato precedentemente. "
            <<"Puoi consultarlo nel file:
"<<nomeFileOutput.substr(10)<<endl<<endl;
        }

        premiInvioPerContinuare();
        break;
    case 2:
        //passo al giorno successivo
        if(!giaCalcolato)
            programmaTUTTEOperazioniGiornaliere();
        nuovoGiorno();

        giaCalcolato = false;
        break;
    case 3:
        //calcolo programma settimanale
        if(!giaCalcolato)
            programmaTUTTEOperazioniGiornaliere();
        nuovoGiorno();

```

```

while(calendario[giornoCorrente].ggsett != 1){
    programmaTUTTEOperazioniGiornaliere();
    nuovoGiorno();
}

cout<<"Programma settimanale calcolato. "
    <<"Puoi consultarlo nel file: "<<nomeFileOutput.substr(10)<<endl<<endl;

premiInvioPerContinuare();
giaCalcolato = false;
break;
case 4:
    //esco dal programma
    cout<<"Arrivederci!"<<endl<<endl;
    system("pause");
    return 0;
default:
    cout << "Scelta non valida, riprova" << endl<<endl;
}
}
}

//————IMPLEMENTAZIONE FUNZIONI————
//-----FUNZIONI DI INIZIALIZZAZIONE (implementazione) -----
void inizializzaAgendaSaleGiornaliera() {
    for(int i = 0; i < NUM_SALE_TUTTE; i++){
        agendaSaleGiornaliera[i].specialitaAssegnata = NESSUNA;
        agendaSaleGiornaliera[i].slotLiberiRimanenti = SLOT_GIORNALIERI_PER_SALA;
        agendaSaleGiornaliera[i].pazOperati = 0;
    }
}
}

```

```

int inizializzaAgendaLetti(){
    string strDaIgnorare;
    uint8_t tmp_totaleLettiDisponibili;

    ifstream file("input\\GO_numLettiPerSpecialita.txt");
    if (!file.is_open()) {
        cerr << "Errore nell'apertura del file GO_numLettiPerSpecialita.txt" << endl;
        exit(EXIT_FAILURE);
    }

    for(int i = 0; i < NUM_SPECIALITA; i++){
        agendaLetti[i].nome = (Specialita)i;
        file >> strDaIgnorare >> tmp_totaleLettiDisponibili;
        agendaLetti[i].totLettiDisponibili = tmp_totaleLettiDisponibili;
        agendaLetti[i].numLettiRimanenti = tmp_totaleLettiDisponibili;
        agendaLetti[i].percLettiRimanenti = (double)100;
    }
    file.close();

    return 0;
}

int costruisciAgenda2024(){
    ifstream file("input\\GO_calendario2024.txt");
    if (!file.is_open()) {
        cerr << "Errore nell'apertura del file GO_calendario2024.txt" << endl;
        exit(EXIT_FAILURE);
    }

    int i = 0, tmp_ggsett;
    while(i<NUM_GIORNI_ANNO &&
file>>calendario[i].data>>tmp_ggsett>>calendario[i].ggfestivo){
        calendario[i].ggsett = static_cast<int>(tmp_ggsett);
        i++;
    }
}

```



```

}
file.close();

if (i < NUM_GIORNI_ANNO) {
    cerr << "Errore nella lettura del file" << endl;
    return -2;
}
else
    return NUM_GIORNI_ANNO;
}

int leggiListaAttesa() {
    unsigned long long int tmp_id;
    string tmp_specialita;
    char tmp_priorita;
    bool tmp_tipo;
    char tmp_durata_operazione;

    ifstream file("input\\GO_listaAttesa.txt");
    if (!file.is_open()) {
        cerr << "Errore nell'apertura del file GO_listaAttesa.txt" << endl;
        exit(EXIT_FAILURE);
    }

    while (file >> tmp_id >> tmp_specialita >> tmp_priorita >> tmp_tipo >>
tmp_durata_operazione){
        if (tmp_tipo)
            pazientiTIPO.push_back({tmp_id, stringToSpecialita(tmp_specialita),
charToPriorita(tmp_priorita), tmp_durata_operazione});
        else
            pazienti.push_back({tmp_id, stringToSpecialita(tmp_specialita),
charToPriorita(tmp_priorita), tmp_durata_operazione});
    }
    file.close();
}

```

```

    pazienti.sort(comparePazienti_id);
    pazientiTIPO.sort(comparePazientiTIPO_id);

    return 0;
}

//-----FUNZIONI PRINCIPALI (implementazione) -----
-----
void nuovoGiorno(){
    giornoCorrente = giornoSuccessivo(giornoCorrente);
    resetSpecialitaOperateOggi();
    inizializzaAgendaSaleGiornaliera();
    numTOTSaleProgrammateOggi = 0;

    if(!calendario[giornoPrecedente()].ggfestivo)
        prioritanoNACercataOggi =
    prossimaPrioritaDaCercare_NON_A(prioritanoNACercataOggi);

    if (calendario[giornoCorrente].ggsett == 1) {
        //è lunedì
        prioritanoNACercataOggi = B;
        inizializzaAgendaLetti();
        nomeFileOutput = "output\agenda_da_" +
    estraiData(calendario[giornoCorrente].data) + "_a_" +
        estraiData(calendario[giornoCorrente + 6].data) + ".txt";

        //se il file esiste già, lo cancello
        cancellaFileOutput();
    }
}

int programmaTUTTEOperazioniGiornaliera(){

```

```

int cont_SalePazTIPO = 0;
int cont_SalePazElez_A = 0;
int cont_SalePazElez_NON_A = 0;
int cont_TOTALESaleAssegnate = 0;

string str = "Giorno "+calendario[giornoCorrente].data+"
"+ggSettToString(calendario[giornoCorrente].ggsett)+
    " - "+prioritaToChar(prioritaNONACercataOggi)+"\n";
scriviSuFileOutput(str, 'a');

if(calendario[giornoCorrente].ggfestivo == true){
    scriviSuFileOutput("\tFESTIVO!\n\n", 'a');
    return 0;
}

//cerco i pazienti TIPO
cont_SalePazTIPO = programmaOperazioniPazTIPO(NUM_SALE_PAZ_TIPO);

if(cont_SalePazTIPO < 2 && !pazientiTIPO.empty()){
    //se ne avevo trovati 0, allora resetto la tabellaRecenti e provo a cercare di nuovo
    //se ne avevo trovati 1, allora resetto nella speranza che tutti i rimanenti non abbiano
    //la stessa specialità del primo trovato, perchè in tal caso non potrei operarli
    resetSpecialitaOperateRecentemente();
    cont_SalePazTIPO += programmaOperazioniPazTIPO(NUM_SALE_PAZ_TIPO -
cont_SalePazTIPO);
}
cont_TOTALESaleAssegnate = cont_SalePazTIPO;

//cerco i pazienti ELEZIONE A
int salePazElezioneDaAssegnare_A = NUM_SALE_TUTTE - cont_SalePazTIPO -
NUM_SALE_PAZ_ELEZ_NON_A; //di solito sarà 8-2-2=4
cont_SalePazElez_A =
programmaOperazioniPazElezione_A(salePazElezioneDaAssegnare_A);
cont_TOTALESaleAssegnate += cont_SalePazElez_A;

```

```

//cerco i pazienti ELEZIONE NON A
    //in teoria le sale da assegnare ai NON A è sempre 2, ma se non sono riuscito ad
    assegnare abbastanza sale tra i
    //pazienti A e i TIPO, allora ne assegno più di 2 ai NON A
    int salePazElezionedaAssegnare_NON_A = NUM_SALE_TUTTE - cont_SalePazTIPO
- cont_SalePazElez_A; //di solito sarà 8-2-4=2
    cont_SalePazElez_NON_A +=
programmaOperazioniPazElezionedaAssegnare_NON_A(salePazElezionedaAssegnare_NON_A);
    cont_TOTALESaleAssegnate += cont_SalePazElez_NON_A;

    if(cont_TOTALESaleAssegnate < NUM_SALE_TUTTE){
        //questo accade se non sono riuscito ad assegnare ai pazienti NON_A tutte le sale
        dedicate a loro, provo allora
        //a cercare di nuovo tra i pazienti A
        cont_TOTALESaleAssegnate +=
programmaOperazioniPazElezionedaAssegnare_A(NUM_SALE_TUTTE -
cont_TOTALESaleAssegnate);
    }

    if(cont_TOTALESaleAssegnate == 0)
        scriviSuFileOutput("\tNON CI SONO ALTRI PAZIENTI DA OPERARE!\n\n", 'a');

    return cont_TOTALESaleAssegnate;
}

//pazienti TIPO
int programmaOperazioniPazTIPO(int numSaleDisponibiliPerOperareTIPO) {
    if (pazientiTIPO.empty())
        return 0;

    int contPazientiTIPO Trovati = 0;
    list<PazienteTIPO>::iterator itr = pazientiTIPO.begin();

```

```

//cerco i due pazienti TIPO da operare
while(itr != pazientiTIPO.end() && contPazientiTIPO Trovati <
numSaleDisponibiliPerOperareTIPO){
    if(checkSpecialitaTIPO((*itr).specialita) == false &&
agendaLetti[(*itr).specialita].numLettiRimanenti > 0){
        assegnaSalaASpecialita((*itr).specialita);
        segnaOperazioneTIPO((*itr).specialita, *itr);
        statisticheFineProgrammazioneSala((*itr).specialita);
        contPazientiTIPO Trovati++;
        itr = pazientiTIPO.erase(itr); //Restituisce un iteratore che punta all'elemento
successivo nella lista dopo quello che è stato cancellato.
    }
    else
        itr++;
}

return contPazientiTIPO Trovati;
}

```

```

//pazienti Elezione A
int programmaOperazioniPazElezione_A(int numSaleDaProgrammare) {
    int cont_saleProgrammate = 0;
    Specialita specDaOperare;
    Priorita prioritaCercata = A;

    if (pazienti.empty())
        return 0;

    while(cont_saleProgrammate < numSaleDaProgrammare){
        specDaOperare = scegliSpecialitaDaOperarePazElezione(prioritaCercata);

        if(specDaOperare == NESSUNA) {
            if(prioritaCercata == D){
                //cout<<"ATTENZIONE: Non ho trovato ulteriori specialita' da

```

```

operare"<<endl;
    return cont_saleProgrammate;
}
else {
    //se non ho trovato nessuna specialità, allora provo con la priorità successiva
    prioritataCercata = (Priorita) (prioritataCercata - 1);
    continue;
}
}

//creo una nuova lista pazienti in cui metto solo quelli che hanno la specialità che
voglio operare
list<Paziente> pazientiConSpecialitaDaOperare;
for(const Paziente &p: pazienti){
    if(p.specialita == specDaOperare)
        pazientiConSpecialitaDaOperare.push_back(p);
}
//ordino la lista sulla base della priorità (ordine DECRESCENTE)
pazientiConSpecialitaDaOperare.sort(comparePazienti_priorita);

//trovo ora i pazienti da operare per quella specialità
list<Paziente>::iterator itr = pazientiConSpecialitaDaOperare.begin();

if(!pazientiConSpecialitaDaOperare.empty()){
    assegnaSalaASpecialita(specDaOperare);
}

while( itr != pazientiConSpecialitaDaOperare.end() &&
agendaSaleGiornaliera[cercaNumSala_Spec(specDaOperare)].slotLiberiRimanenti > 0
&&
agendaLetti[cercaNumSala_Spec(specDaOperare)].numLettiRimanenti > 0){

```

```

if(agendaSaleGiornaliera[cercaNumSala_Spec(specDaOperare)].slotLiberiRimanenti >=
durataInterventoPaziente(*itr)){
    //se ho ancora letti disponibili e slot sufficienti, allora opero il paziente
    segnaOperazioneElezione(specDaOperare, *itr);
    eliminaPazElezioneDaListaAttesa_id((*itr).id);
    itr = pazientiConSpecialitaDaOperare.erase(itr);
}
else
    itr++;

//ATTENZIONE: può capitare che io sia nel loop con uno slot libero rimasto ma ho
solo pazienti con durata
//operazione lunga, in tal caso non rimango bloccato nel loop perché uscirò una
volta scorsa tutta la lista
}

statisticheFineProgrammazioneSala(specDaOperare);
cont_saleProgrammate++;
}

return cont_saleProgrammate;
}

//pazienti Elezione NON A
Priorita prossimaPrioritaDaCercare_NON_A(Priorita prioritaCercata){
    switch (prioritaCercata) {
        case B: return C;
        case C: return D;
        case D: return B;
        default: cerr << "ERRORE: PrioritaCercata non valida" << endl;
    }
    exit(EXIT_FAILURE);
}
}

```

```

int programmaOperazioniPazElezioni_NON_A(int numSaleDaProgrammare){
    int cont_saleProgrammate = 0;
    Specialita specDaOperare;
    Priorita prioCercata = prioritanoNACercataOggi;
    int cont_tentativiRicerca = 0;

    if (pazienti.empty())
        return 0;

    while(cont_saleProgrammate < numSaleDaProgrammare){
        //scelgo la specialità da operare sulla base della percentuale di letti rimanenti e della
        //priorità non A che si
        //vuole operare oggi
        specDaOperare = scegliSpecialitaDaOperarePazElezioni(prioCercata);
        cont_tentativiRicerca++;

        //se non ho trovato nessuna specialità, magari perché non ho pazienti con quella priorità
        //che devono essere //operati per specialità diverse da quelle già programmate oggi, allora
        //provo con la priorità successiva
        if(specDaOperare == NESSUNA) {
            //può capitare (in caso di scarsità di pazienti) che dopo aver cercato fra tutte le
            //priorità minori di A, io
            //non abbia trovato nessuna specialità da operare, in tal caso esco dalla funzione
            if(cont_tentativiRicerca >= 3){
                return cont_saleProgrammate;
            }
            else{
                prioCercata = prossimaPrioritaDaCercare_NON_A(prioCercata);
                continue;
            }
        }
    }

    cont_tentativiRicerca = 0;
    assegnaSalaASpecialita(specDaOperare);
}

```



```

Priorita prioDaCercareNelLoop = prioCercata;

//decisa la specialità da operare, cerco i pazienti da operare

while(agendaSaleGiornaliera[cercaNumSala_Spec(specDaOperare)].slotLiberiRimanenti
> 0 && cont_tentativiRicerca < 3) {
    //creo una nuova lista pazienti in cui metto solo quelli che hanno la specialità che
voglio operare
    //e che abbiano la prima priorità cercata
    list<Paziente> lista_SpecEPrio =
costruisciListaPazElezione_SpecEPrio(specDaOperare, prioDaCercareNelLoop);
    cont_tentativiRicerca++;

    if (lista_SpecEPrio.empty()){
        prioDaCercareNelLoop =
prossimaPrioritaDaCercare_NON_A(prioDaCercareNelLoop);
        continue;
    }

    //trovo ora i pazienti da operare per quella specialità con la priorità cercata
    list<Paziente>::iterator itr = lista_SpecEPrio.begin();

    while (itr != lista_SpecEPrio.end() &&

agendaSaleGiornaliera[cercaNumSala_Spec(specDaOperare)].slotLiberiRimanenti > 0
&&
        agendaLetti[cercaNumSala_Spec(specDaOperare)].numLettiRimanenti > 0) {

        if
(agendaSaleGiornaliera[cercaNumSala_Spec(specDaOperare)].slotLiberiRimanenti >=
durataInterventoPaziente(*itr)) {
            //se ho ancora letti disponibili e slot sufficienti, allora opero il paziente
segnaOperazioneElezione(specDaOperare, *itr);
            eliminaPazElezioneDaListaAttesa_id((*itr).id);

```

```

        itr = lista_SpecEPrio.erase(itr);
    }
    else
        itr++;

//ATTENZIONE: può capitare che io sia nel loop con uno slot libero rimasto ma ho solo
//pazienti con //durata operazione lunga, in tal caso non rimango bloccato nel loop perché
//uscirò una volta scorsa //tutta la lista
    }

    //se sono uscito dal loop perché ho finito i pazienti con la priorità cercata, ma ho
    //ancora slot liberi,
    //allora provo con la priorità successiva

    if(agendaSaleGiornaliera[cercaNumSala_Spec(specDaOperare)].slotLiberiRimanenti >
    0){
        prioDaCercareNelLoop =
        prossimaPrioritaDaCercare_NON_A(prioDaCercareNelLoop);
    }
}

    statisticheFineProgrammazioneSala(specDaOperare);
    cont_tentativiRicerca = 0;
    cont_saleProgrammate++;
}

    return cont_saleProgrammate;
}

//in comune tra i pazienti Elezione A e NON A
Specialita scegliSpecialitaDaOperarePazElezione(Priorita prioritaCercata){
    //cerco la specialità che ha più posti liberi
    Specialita specDaOperare = NESSUNA;
}

```

```

//mi faccio una copia di agendaLetti
Letti copia[NUM_SPECIALITA];
for(int i = 0; i < NUM_SPECIALITA; i++){
    copia[i] = agendaLetti[i];
}

//ordino la copia_agendaLetti sulla base di percLettiRimanenti (ordine
DESCRESCENTE)
for(int i = 0; i < NUM_SPECIALITA; i++){
    for(int j = i+1; j < NUM_SPECIALITA; j++){
        if(copia[i].percLettiRimanenti < copia[j].percLettiRimanenti ||
            (copia[i].percLettiRimanenti == copia[j].percLettiRimanenti &&
                posizionePrimoPazElezione_SpecEPrio(copia[i].nome, prioritataCercata) >
                posizionePrimoPazElezione_SpecEPrio(copia[j].nome, prioritataCercata))){
            Letti tmp = copia[i];
            copia[i] = copia[j];
            copia[j] = tmp;
        }
    }
}

int index = 0;
while(index < NUM_SPECIALITA) {
    Specialita specialitaCandidata = copia[index].nome;
    //controllo se non l'ho già operata oggi
    if (checkSpecialitaElezione(specialitaCandidata) == false &&
        checkPresenzaPazElezione_SpecEPrio(specialitaCandidata, prioritataCercata) ==
        true &&
        agendaLetti[specialitaCandidata].numLettiRimanenti > 0) {
        specDaOperare = specialitaCandidata;
        break;
    }
    else

```

```

        index++;
    }
    return specDaOperare;
}

```

```

//-----FUNZIONI AUSILIARIE (implementazione)-----
list<Paziente> costruisciListaPazElezione_SpecEPrio(Specialita specDaOperare, Priorita
prioritaCercata) {
    //creo una nuova lista pazienti in cui metto solo quelli che hanno la specialità che voglio
operare con la priorità cercata
    list<Paziente> lista;
    for (const Paziente &p: pazienti) {
        if (p.specialita == specDaOperare && p.priorita == prioritaCercata)
            lista.push_back(p);
    }

    return lista;
}

```

```

void assegnaSalaASpecialita(Specialita s){
    if(numTOTSaleProgrammateOggi >= NUM_SALE_TUTTE){
        cerr<<"ERRORE: Sale esaurite"<<endl; exit(EXIT_FAILURE);}
    else if( (agendaSaleGiornaliera[numTOTSaleProgrammateOggi].specialitaAssegnata
!= NESSUNA)){
        cerr<<"ERRORE: Sala gia' occupata"<<endl; exit(EXIT_FAILURE);}
    else if(agendaSaleGiornaliera[numTOTSaleProgrammateOggi].slotLiberiRimanenti <
SLOT_GIORNALIERI_PER_SALA){
        cerr<<"ERRORE: La sala che si voleva assegnare ha gia' dei posti
occupati"<<endl; exit(EXIT_FAILURE);}

    agendaSaleGiornaliera[numTOTSaleProgrammateOggi].specialitaAssegnata = s;
    numTOTSaleProgrammateOggi++;
}

```

```

    ostream oss;
    oss<<"\tSala " <<numTOTSaleProgrammateOggi<<" - ("<<s<<"
"<<specialitaToString(s)<<endl;
    scriviSuFileOutput(oss.str(), 'a');
}

void statisticheFineProgrammazioneSala(Specialita s){
    int numSala = cercaNumSala_Spec(s);
    ostream oss;
    oss<<"\t\t-----" <<endl;
    oss<<"\t\t numPaz: " <<static_cast<int>(agendaSaleGiornaliera[numSala].pazOperati)
    <<" , slotLiberiRimanenti:
"<<static_cast<int>(agendaSaleGiornaliera[numSala].slotLiberiRimanenti)<<endl<<en
dl;
    scriviSuFileOutput(oss.str(), 'a');
}

int cercaNumSala_Spec(Specialita x){
    for(int i = 0; i < NUM_SALE_TUTTE; i++){
        if(agendaSaleGiornaliera[i].specialitaAssegnata == x){
            return i;
        }
    }
}

cerr<<"ERRORE: La Specialita' " <<x<<" non ha una sala assegnata!" <<endl;
exit(EXIT_FAILURE);
}

bool checkPresenzaPazElezione_SpecEPrio(Specialita x, Priorita y){
    bool trovato = false;
    for(const Paziente &p: pazienti){
        if(p.specialita == x && p.priorita == y){
            trovato = true;
            break;
        }
    }
}

```

```

    }
}

return trovato;
}

unsigned int posizionePrimoPazElezione_SpecEPrio(Specialita s, Priorita p){
    //scorro la lista dei pazienti e restituisco la posizione del primo paziente che ha la
    specialità s e la priorità p
    int pos = 0;
    for(const Paziente &paz: pazienti){
        if(paz.specialita == s && paz.priorita == p)
            return pos;
        else
            pos++;
    }

    return numeric_limits<unsigned int>::max();
}

void eliminaPazElezioneDaListaAttesa_id(unsigned long long int id){
    list<Paziente>::iterator itr = pazienti.begin();
    while(itr != pazienti.end()){
        if((*itr).id == id){
            pazienti.erase(itr);
            return;
        }
        else
            itr++;
    }

    cerr<<"ERRORE: Paziente da eliminare NON trovato"<<endl;
    exit(EXIT_FAILURE);
}

```

```

void segnaOperazioneTIPO(Specialita x, const PazienteTIPO &p){
    SpecialitaOperateRecentemente[x] = true;
    specialitaOperateOggi[x] = true;

    //aggiorno agendaSaleGiornaliera -> diventa tutta piena perchè pazienteTIPO
    agendaSaleGiornaliera[cercaNumSala_Spec(x)].pazOperati = 1;
    agendaSaleGiornaliera[cercaNumSala_Spec(x)].slotLiberiRimanenti=0;

    //aggiorno agendaLetti
    agendaLetti[x].numLettiRimanenti--;
    agendaLetti[x].percLettiRimanenti = (double)agendaLetti[x].numLettiRimanenti /
(double)agendaLetti[x].totLettiDisponibili * 100;

    //output su file
    ostream oss;
    oss << "\t\tpaz: " << setw(16) << setfill('0') << p.id
        << ", prio: " << prioritatoChar(p.priorita)
        << ", durata: " << p.durata_operazione << '(' << SLOT_OPER_TIPO << ')' << " -
pazTIPO" << endl;
    scriviSuFileOutput(oss.str(), 'a');

    //conto quante specialità sono già state operate recentemente
    int cont_specOperateRecentemente = 0;
    for(int i = 0; i < NUM_SPECIALITA; i++){
        if(SpecialitaOperateRecentemente[i] == true)
            cont_specOperateRecentemente++;
    }

    //se sono già state operate 10 specialità recentemente, allora resetto la tabella
    if(cont_specOperateRecentemente >= 10){
        for(int i = 0; i < NUM_SPECIALITA; i++)
            SpecialitaOperateRecentemente[i] = false;
    }
}

```

```
}
```

```
void segnaOperazioneElezione(Specialita x, const Paziente &p){  
    specialitaOperateOggi[x] = true;  
  
    agendaSaleGiornaliera[cercaNumSala_Spec(x)].pazOperati++;  
    agendaSaleGiornaliera[cercaNumSala_Spec(x)].slotLiberiRimanenti -=  
durataInterventoPaziente(p);  
  
    //aggiorno agendaLetti  
    agendaLetti[x].numLettiRimanenti--;  
    agendaLetti[x].percLettiRimanenti = (double)agendaLetti[x].numLettiRimanenti /  
(double)agendaLetti[x].totLettiDisponibili * 100;  
  
    //output su file  
    ostringstream oss;  
    oss << "\t\tpaz: " << setw(16) << setfill('0') << p.id  
        << ", prio: " << prioritatoChar(p.priorita)  
        << ", durata: " << p.durata_operazione << '(' <<  
static_cast<int>(durataInterventoPaziente(p)) << ')' << endl;  
    scriviSuFileOutput(oss.str(), 'a');  
}
```

```
bool checkSpecialitaTIPO(Specialita x){  
    //return true se la specialità è già stata operata OGGI o recentemente, false altrimenti  
    if(SpecialitaOperateRecentemente[x] == true ||  
        specialitaOperateOggi[x] == true)  
        return true;  
    else  
        return false;  
}
```

```
bool checkSpecialitaElezione(Specialita x){  
    //return true se la specialità è già stata operata OGGI, false altrimenti
```



```

    if(specialitaOperateOggi[x] == true)
        return true;
    else
        return false;
}

void resetSpecialitaOperateRecentemente(){
    for(int i = 0; i < NUM_SPECIALITA; i++){
        SpecialitaOperateRecentemente[i] = false;
    }
}

void resetSpecialitaOperateOggi(){
    for(int i = 0; i < NUM_SPECIALITA; i++){
        specialitaOperateOggi[i] = false;
    }
}

void showMenu(){
    string caption_0 = "***** MENU INIZIALE
*****";
    string caption_1 = "Data di oggi: "+calendario[giornoCorrente].data+",
"+ggSettToString(calendario[giornoCorrente].ggsett);
    ostreamstream caption_2;
    caption_2<<"Digita uno dei seguenti numeri per selezionare una delle opzioni
desiderate:\n"
        <<"\t1) Visualizzare la programmazione operatoria in data odierna\n"
        <<"\t2) Passa al giorno seguente\n"
        <<"\t3) Visualizzare la programmazione operatoria nella settimana corrente\n"
        <<"\t4) Exit";
    string caption_3 =
    "*****";

    cout << caption_0 << endl << caption_1 << endl << endl << caption_2.str() <<

```

```
endl << caption_3 << endl;
    cout << "Opzione scelta: ";
}
```

```
void scriviSuFileOutput(const string &str, const char &mode){
```

```
    // mode = 'a' (append) oppure 'w' (write)
```

```
    ofstream file; // Dichiarazione della variabile file
```

```
    switch (mode) {
```

```
        case 'a':
```

```
            file.open(nomeFileOutput, ios::app); // Apertura per append
```

```
            break;
```

```
        case 'w':
```

```
            file.open(nomeFileOutput); // Apertura standard (truncation di default)
```

```
            break;
```

```
        default:
```

```
            cerr << "ERRORE: mode non valido" << endl;
```

```
            exit(EXIT_FAILURE);
```

```
    }
```

```
    if (!file.is_open()) {
```

```
        cerr << "Errore nell'apertura del file in scrittura: " << nomeFileOutput << endl;
```

```
        exit(EXIT_FAILURE);
```

```
    }
```

```
    file << str;
```

```
    file.close();
```

```
}
```

```
void cancellaFileOutput(){
```

```
    ifstream file(nomeFileOutput);
```

```
    if (file.good()) {
```

```
        file.close();
```

```
        remove(nomeFileOutput.c_str());
```

```
}  
}
```

```
unsigned int riceviNumOperazione() {  
    unsigned int scelta;  
  
    while (true) {  
        if ((cin >> scelta) && (scelta <= 4)) {  
            // Input valido e nel range specificato  
            break;  
        } else {  
            // Input non valido o fuori dal range  
            cout << "Per favore, inserisci un numero compreso tra 0 e 4." << endl << endl;  
            cout << "Opzione scelta: ";  
  
            // Pulisci il flag di errore di cin  
            cin.clear();  
  
            // Ignora l'input non valido fino al prossimo newline  
            cin.ignore(numeric_limits<streamsize>::max(), '\n');  
        }  
    }  
  
    // Svuota il buffer di sistema (flush) prima di terminare  
    // così da consumare eventuali caratteri newline rimasti  
    cin.ignore(numeric_limits<streamsize>::max(), '\n');  
  
    return scelta;  
}
```

```
void premiInvioPerContinuare(){  
    cout<<"Premi INVIO per continuare\n";  
  
    //uso la getline così da consumare completamente il buffer di sistema
```

```

//compreso il carattere newline finale
string discardedInput;
getline(cin, discardedInput);
}

///-----FUNZIONI DI UTILITÀ (implementazione) -----
Specialita stringToSpecialita(const string &str){
    if(str == "NESSUNA") return NESSUNA;
    else if(str == "BU") return BU;
    else if(str == "CH_GEN") return CH_GEN;
    else if(str == "CH_MXF") return CH_MXF;
    else if(str == "CH_ODONT") return CH_ODONT;
    else if(str == "CH_OCU") return CH_OCU;
    else if(str == "CH_ORL") return CH_ORL;
    else if(str == "CH_ORTOP") return CH_ORTOP;
    else if(str == "CH_PED") return CH_PED;
    else if(str == "CH_PLA") return CH_PLA;
    else if(str == "CH_URO") return CH_URO;
    else if(str == "CH_VASC") return CH_VASC;
    else if(str == "EBUS") return EBUS;
    else if(str == "ENDO_DIG") return ENDO_DIG;
    else if(str == "PI_DIAB") return PI_DIAB;
    else if(str == "TARF") return TARF;
    else if(str == "T_ANT") return T_ANT;

    cerr<<"ERRORE: Specialita non valida"<<endl; exit(EXIT_FAILURE);
}

string specialitaToString(const Specialita &s){
    if(s==-1) return "NESSUNA";
    else if(s==0) return "BU";
    else if(s==1) return "CH_GEN";
    else if(s==2) return "CH_MXF";
    else if(s==3) return "CH_ODONT";
}

```

```

else if(s==4) return "CH_OCU";
else if(s==5) return "CH_ORL";
else if(s==6) return "CH_ORTOP";
else if(s==7) return "CH_PED";
else if(s==8) return "CH_PLA";
else if(s==9) return "CH_URO";
else if(s==10) return "CH_VASC";
else if(s==11) return "EBUS";
else if(s==12) return "ENDO_DIG";
else if(s==13) return "PI_DIAB";
else if(s==14) return "TARF";
else if(s==15) return "T_ANT";

cerr<<"ERRORE: Specialita non valida"<<endl; exit(EXIT_FAILURE);
}

```

```

Priorita charToPriorita(const char &c){

```

```

    switch(c) {
        case 'A': return A;
        case 'B': return B;
        case 'C': return C;
        case 'D': return D;
        default: cerr << "ERRORE: Priorita non valida" << endl; exit(EXIT_FAILURE);
    }
}

```

```

char prioritaToChar(const Priorita& p){

```

```

    switch(p) {
        case A: return 'A';
        case B: return 'B';
        case C: return 'C';
        case D: return 'D';
        default: cerr << "ERRORE: Priorita non valida" << endl; exit(EXIT_FAILURE);
    }
}

```

```
}
```

```
string ggSettToString(const uint8_t& ggSett){  
    switch (ggSett) {  
        case 1: return "Lunedì";  
        case 2: return "Martedì";  
        case 3: return "Mercoledì";  
        case 4: return "Giovedì";  
        case 5: return "Venerdì";  
        case 6: return "Sabato";  
        case 7: return "Domenica";  
        default: cerr << "ERRORE: ggSett non valido" << endl; exit(EXIT_FAILURE);  
    }  
}
```

```
bool comparePazientiTIPO_id(const PazienteTIPO &a, const PazienteTIPO &b) {  
    //ordine CRESCENTE  
    return a.id < b.id;  
}
```

```
bool comparePazienti_id(const Paziente &a, const Paziente &b) {  
    //ordine CRESCENTE  
    return a.id < b.id;  
}
```

```
bool comparePazienti_priorita(const Paziente& a, const Paziente& b) {  
    //ordine DECRESCENTE  
    return a.priorita > b.priorita;  
}
```

```
uint8_t durataInterventoPaziente(const Paziente &p){  
    if(p.durata_operazione == 'T') return SLOT_OPER_TIPO;  
    else if(p.durata_operazione == 'L') return SLOT_OPER_LUNGA;  
    else if(p.durata_operazione == 'C') return SLOT_OPER_CORTA;
```

```

else if(p.durata_operazione == 'E') return SLOT_OPER_LOCALE;

cerr<<"ERRORE: Durata operazione non valida"<<endl; exit(EXIT_FAILURE);
}

short int giornoPrecedente(){
    if(giornoCorrente == 0)
        return NUM_GIORNI_ANNO - 1;
    else
        return giornoCorrente - 1;
}

short int giornoSuccessivo(const short int &g){
    if(g == NUM_GIORNI_ANNO - 1)
        return 0;
    else
        return g + 1;
}

string estraiData(const string& dataCompleta) {
    if (dataCompleta.length() >= 5) {
        // Estrai la sottostringa a partire dall'indice 5 (che corrisponde al carattere dopo
        "AAAA-")
        return dataCompleta.substr(5);
    }
    return dataCompleta; // Restituisci la stringa originale se non è abbastanza lunga
}

```

APPENDICE B

Di seguito vengono riportati i dati utilizzati per generare le istanze di Mosel, prese in esame per l'implementazione del modello tramite l'algoritmo.

```
model knapsack
uses "mmxprs", "mmive", "mmsystem"

declarations
x: array(1..500, 1..8, 1..30) of mpvar !10 giorni 20 slot
y: array(1..16, 1..8, 1..30) of mpvar
w,p,s : array(1..500) of integer
numpaz, W, sala, slot, sp: integer
end-declarations

setparam("XPRS_MAXTIME",120)

sala:=8
slot:=20
sp:=16

fopen("knapsack.txt",F_INPUT)
read(numpaz)
read(W)

forall(i in 1..numpaz)do
    read(s(i))
end-do

forall(i in 1..numpaz)do
    read(w(i))
```



```

end-do

forall(i in 1..numpaz)do
    read(p(i))
end-do

writeln("Numero pazienti nella lista d'attesa: ", numpaz)
writeln("Capacita massima della sala operatoria: ", W, " minuti")
writeln("\n")

forall(i in 1..numpaz) do
    writeln("Specialita chirurgica paziente ", i, ": ", s(i))
    writeln("Durata intervento paziente ", i, ": ", w(i))
    writeln("Peso del paziente sulla base della priorit  ", i, ": ", p(i))
    writeln("\n")
end-do

fclose(F_INPUT)

forall(i in 1..numpaz, j in 1..sala, k in 1..slot) x(i,j,k) is_binary

forall(l in 1..sp, j in 1..sala, k in 1..slot) y(l,j,k) is_binary

obj:= sum(i in 1..numpaz, j in 1..sala, k in 1..slot) w(i)*p(i)*x(i,j,k)

forall (i in 1..numpaz, j in 1..sala, k in 1..slot) do
    x(i,j,k)<= y(s(i),j,k)
end-do

forall (j in 1..sala, k in 1..slot) do
    ctrl1(j,k):= sum(i in 1..numpaz) w(i)*x(i,j,k)<= W    !W=720
end-do

```

```

forall (j in 1..sala, k in 1..slot) do
    !Non piu di due pazienti in TIPO
    ctrl2(j,k):=sum(i in 1..numpaz | p(i)=100) x(i,j,k)<=1
end-do

forall (j in 1..sala, k in 1..slot) do
    !Almeno 8 pazienti di priorit? B, C, D -> 8 in una sala AL, 4 nell'altra C
    ctrl3(j,k):=sum(i in 1..numpaz | p(i)=5) x(i,j,k)>=2
end-do

forall (j in 1..sala, k in 1..slot) do
    !Numero pazienti operabili al giorno su 8 sale stimando di operare 24 pazienti di
priorita A
    ctrl4(j,k):=sum(i in 1..numpaz) x(i,j,k) <=340
end-do

!Ogni intervento può essere fatto in un'unica sala
forall (i in 1..numpaz) do
    sum(j in 1..sala, k in 1..slot)x(i,j,k) <=1
end-do

forall (l in 1..sp,k in 1..slot) do
    sum(j in 1..sala)y(l,j,k)<=1
end-do

maximize(obj)

writeln("Il punteggio massimo del lasso di tempo considerato: ",getobjval)

exportprob("FILE1.lp")
end-model

```

DATI INPUT MOSEL

Gli spazi tra le righe sono stati volutamente lasciati al fine di rendere più leggibile il file di input. Ovviamente ogni riga, esclusa la prima, contiene 500 numeri differenti: uno per ogni paziente.

500 360

13 11 7 12 9 1 5 16 12 7 4 10 15 9 13 12 11 9 11 14 16 8 10 5 11 10 4 12 2 11 9 16 1 5 10 3
12 11 9 16 2 12 14 13 6 7 11 11 3 4 16 6 13 16 15 8 1 12 5 5 15 6 11 1 13 7 15 8 16 2 6 3 11
9 3 1 7 9 10 3 5 9 4 14 10 8 6 7 2 10 14 15 12 8 9 12 13 8 10 1 3 8 7 3 2 5 11 10 3 12 12 4 6
6 4 16 5 8 4 16 14 3 15 16 13 9 5 12 16 13 5 9 13 5 7 16 9 7 4 3 2 11 11 16 10 14 6 10 11 5
10 8 7 8 6 12 12 7 5 5 2 2 1 5 2 2 3 12 3 2 6 7 3 11 3 7 14 14 8 6 7 9 8 15 8 15 6 14 15 7 12
14 5 4 3 15 10 10 5 7 13 4 5 10 2 16 2 13 7 8 11 9 16 4 1 1 1 15 10 4 9 13 3 3 1 9 16 5 3 7 3
7 13 6 14 2 9 8 16 13 7 13 13 13 8 2 12 8 14 4 3 16 6 7 16 1 5 6 10 4 6 16 11 16 12 7 11 1 8
2 15 7 8 8 13 8 5 7 7 5 2 5 5 9 10 11 11 5 11 1 11 11 3 1 1 1 1 1 14 8 7 6 2 2 2 13 4 15 11 6
6 7 7 4 7 5 6 6 6 7 4 12 6 13 2 2 3 4 11 2 10 3 3 6 3 10 9 1 10 2 6 2 2 2 6 1 1 1 7 1 1 10 6 5
7 10 10 1 5 8 9 9 8 4 4 11 9 10 14 10 4 1 12 7 6 10 2 11 11 9 2 4 6 2 7 2 12 5 13 11 3 1 10 1
14 7 2 5 7 4 3 7 8 8 12 8 2 12 7 11 4 1 1 5 1 1 1 15 11 7 11 2 3 2 2 5 1 2 9 6 3 5 7 1 10 3 2 7
3 7 8 8 9 11 9 11 11 4 2 11 2 1 1 1 6 1 3 11 12 3 1 10 5 4 1 2 2 2 14 15 4 2 3 8 1 4 7 13 13 4
4 3 4 9 2 7 3 7 6 11 11 7 1 11 2 7 6 7 2 1

360 720 180 360 360 90 90 360 180 360 360 180 180 360 360 180 90 720 180 180 180 180
360 90 360 180 180 360 180 180 180 360 90 360 180 720 90 360 360 720 360 360 90 90
720 360 360 180 360 360 180 180 360 180 360 360 360 180 90 90 360 90 720 720 360 720
360 360 180 180 720 360 180 180 180 720 720 720 720 360 180 360 360 180 360 180 180
360 360 180 360 180 360 360 90 180 360 90 720 90 360 180 360 180 360 90 180 360 720
180 180 360 720 90 360 360 180 360 90 90 90 360 90 90 90 180 180 180 180 180 360 360
90 360 720 90 180 360 90 360 90 180 360 90 180 180 90 360 180 180 720 720 720 360 360
360 90 360 180 90 90 360 360 180 90 360 360 90 180 360 180 180 360 360 180 360 180
180 90 180 90 360 180 180 720 180 360 180 90 180 360 360 180 360 360 360 180 180 360

180 360 90 90 180 720 180 180 360 180 720 90 180 360 180 90 90 720 180 360 360 180
180 90 720 360 360 90 180 180 360 180 360 180 90 360 90 720 180 90 360 360 90 360 360
360 360 180 720 90 90 180 360 90 360 90 90 180 90 360 90 360 180 360 180 360 180 360
360 720 360 180 720 180 180 90 90 360 360 180 90 360 180 180 90 360 180 720 360 180
90 720 180 360 90 360 90 180 90 360 360 90 360 720 180 90 360 180 180 90 180 180 180
180 180 180 360 180 90 180 90 360 360 180 360 180 90 360 90 360 720 90 360 180 360
180 180 180 720 180 720 180 360 180 360 90 90 360 180 90 180 360 180 180 90 180 360
360 360 90 90 360 90 360 180 90 360 720 90 720 360 90 90 180 360 360 360 180 180 720
90 180 90 360 720 180 180 90 90 180 90 720 90 360 90 180 720 720 90 180 180 90 180
360 360 180 90 180 180 180 720 90 180 180 180 360 360 360 180 180 180 180 180 90 90
360 180 180 90 360 90 180 90 360 180 360 360 360 360 720 180 180 360 360 180 180 90
180 90 180 720 90 180 90 720 720 90 180 180 90 180 180 180 90 180 90 720 180 360 180
180 360 180 90 90 90 180 360 180 180 90 180 720 360 90 90 90 360 180 360 180 360 180
180 360 360 90 180 90 90 360

1 100 1 1 10 1 10 1 1 1 10 10 10 10 10 10 1 100 10 1 1 10 1 1 10 10 10 10 10 1 1 1 10 1
100 1 1 10 100 1 1 1 1 100 10 10 10 1 1 10 1 10 10 1 10 1 10 1 10 1 1 100 100 10 100 1 1 1
10 100 10 10 10 1 100 100 100 100 10 10 10 10 1 1 1 1 1 10 1 10 10 10 1 1 1 1 100 10 1
1 10 1 10 10 10 1 100 10 1 10 100 1 10 10 1 10 1 1 1 1 10 1 1 1 10 10 10 1 1 1 10 10 100 10
1 10 1 10 1 10 10 1 1 1 10 10 10 10 100 100 100 10 10 10 1 10 10 10 1 10 10 1 1 1 1 1 1
1 10 10 10 1 1 10 10 1 1 10 1 1 1 100 1 1 1 10 10 10 1 10 1 1 1 1 10 1 1 10 10 1 100 1 1 1
1 100 1 10 10 10 1 1 100 10 10 10 1 10 1 100 1 1 10 1 1 10 10 1 1 1 1 100 10 1 10 10 1 10
10 10 10 10 100 1 1 10 10 1 10 10 10 10 1 10 1 10 10 10 10 10 1 10 10 100 1 10 100 1 10 1
1 1 10 10 1 10 1 1 10 10 1 100 10 1 1 100 10 10 1 10 1 1 1 10 10 1 1 100 1 1 10 1 10 1 1 1
1 1 10 10 10 10 1 10 1 10 10 10 10 10 1 1 10 10 100 1 1 1 10 1 10 10 100 10 100 10 10 10
10 1 10 1 1 10 1 10 1 10 1 10 10 10 10 10 1 10 1 10 10 1 10 100 1 100 10 10 10 10 10 10
10 10 100 10 10 1 10 100 10 10 10 1 10 10 100 1 10 10 10 100 100 1 10 10 10 10 10 10 10
10 10 10 10 100 10 10 1 1 1 10 10 1 10 10 10 10 1 10 1 1 10 10 10 10 1 10 10 10 10 10
10 100 10 10 10 10 10 1 10 10 10 10 100 10 10 10 100 100 1 1 1 1 10 10 1 1 1 1 100 10 1 1
10 10 10 1 10 10 10 10 1 1 1 1 100 10 1 10 10 10 10 1 10 1 10 10 10 10 10 10 10

Bibliografia/Sitografia

- [1] <https://lamagistratura.it/commentario/lart-32-della-costituzione/>
- [2] <https://www.salute.gov.it/portale/lea/dettaglioContenutiLea.jsp?lingua=italiano&id=5073&area=Lea&menu=vuoto>
- [3] https://www.quotidianosanita.it/cronache/articolo.php?articolo_id=103278
- [4] [https://it.wikipedia.org/wiki/Servizio_sanitario_nazionale_\(Italia\)](https://it.wikipedia.org/wiki/Servizio_sanitario_nazionale_(Italia))
- [5] <https://www.governo.it/it/costituzione-italiana/parte-seconda-ordinamento-della-repubblica/titolo-v-le-regione-province-e-i>
- [6] <https://www.gazzettaufficiale.it/eli/id/1992/10/31/092G0463/sg>
- [7] <https://www.altalex.com/documents/news/2012/09/14/decreto-balduzzi-sulla-sanita-in-14-punti>
- [8] <https://leg16.camera.it/561?appro=853>
- [9] <https://alpi.agenas.it/>
- [10] <https://www.parlamento.it/parlam/leggi/071201.htm>
- [11] https://www.asl1sassari.it/wp-content/uploads/2024/03/PDEL_2024_0000304_Regolamento_P.A._ULTIMA_VE_RSIONE_del_19_febbraio_2024-n.-308.pdf
- [12] <https://www.sardegna salute.it/assistenza/>
- [13] <https://www.siems.it/mmg.html>
- [14] <https://www.aousassari.it/>
- [15] <https://www.nurse24.it/infermiere/leggi-normative/legge-833-78-istituzione-servizio-sanitario-nazionale.html>
- [16] https://www.aousassari.it/documenti/11_193_20171010165426.pdf
- [17] https://health.ec.europa.eu/cross-border-healthcare/overview_it
- [18] <https://www.gomrc.it/reparto/uosd-terapia-intensiva-post-operatoria-t-i-p-o/>
- [19] <https://www.nurse24.it/specializzazioni/management-universita-area-forense/rapporto-infermieri-pazienti-la-determinazione-dell-organico.html>
- [20] <http://www.studiolegalecarozza.it/news/ferie-a-titolo-di-congedo-ordinario-e-per-rischio-radiologico.htm>

- [21] <https://cyberplan.it/il-rough-cut-capacity-planning-o-rccp/>
- [22] Zhu, S., Fan, W., Yang, S. et al. Operating room planning and surgical case scheduling: a review of literature. *J Comb Optim* 37, 757–805, 2019.
- [23] Tadei R., Della Croce F.; *Elementi di ricerca operativa*, Esculapio, 2010 – seconda edizione.
- [24] https://www.math.unipd.it/~luigi/courses/ricop0809/ro_13.m04.BeB.01.pdf
- [25] https://www.diag.uniroma1.it/~bruni/files/OC_4Branch&Bound.pdf
- [26] <https://core.ac.uk/download/pdf/11194226.pdf>
- [27] Z H Przasnyski, *Operating room scheduling. A literature review*, AORN Journal, 44 (1), 67-79, 1986.
- [28] <http://groups.di.unipi.it/optimize/Courses/RO2IG/aa1415/3-bb.pdf>
- [29] <https://www.fico.com/fico-xpress-optimization/docs/latest/overview.html>
- [30] https://www.fico.com/fico-xpress-optimization/docs/dms2019-03/solver/optimizer/HTML/chapter4_sec_section4003.html
- [31] <https://magistri.altervista.org/SISTEMI/terza/Guidaallinguaggioc.pdf>
- [32] <https://www.html.it/guide/guida-c2/>
- [33] <https://www.treccani.it/>
- [34] B Zhang, P Murali, M M Dessouky & D Belson, *A mixed integer programming approach for allocating operating room capacity*, Journal of the Operational Research Society, 60:5, 663-673, 2009.
- [35] Vincent Augusto, Xiaolan Xie, Viviana Pedromo, *Operating theatre scheduling with patient recovery in both operating rooms and recovery beds*, Computers and Industrial Engineering, 58:2, 231-238, 2010.
- [36] Line Ravnskjær Kroer, Karoline Foverskov, Charlotte Vilhelmsen, Aske Skouboe Hansen, Jesper Larsen; *Planning and scheduling operating rooms for elective and emergency surgeries with uncertain duration*, Volume 19, 2018.
- [37] Kieran Walshe, Judith Smith; *Healthcare Management*, McGraw Hill, 2016 - terza edizione.
- [38] Moon, Weon-Hee; *Development and evaluation of NRMIS (Nursing Resources Management Information System) for managing healthcare resources*, Technology and Health Care, 2019.
- [39] Göras, C., Nilsson, U., Ekstedt, M. et al. *Managing complexity in the operating room: a group interview study*. BMC Health Serv Res 20, 440 (2020).

<https://doi.org/10.1186/s12913-020-05192-8>

- [40] Duma, D., Aringhieri, *The Real Time Management of Operating Rooms*. In: Kahraman, C., Topcu, Y. (eds) *Operations Research Applications in Health Care Management*. International Series in Operations Research & Management Science, vol 262. Springer, 2018.
- [41] <https://cerinozegna.it/camera-di-biocontenimento-a-pressione-negativa/>