



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di laurea in Informatica

Analisi di una rete logistica per lo scambio
interbibliotecario

Relatore: Prof. Alberto Ceselli

Correlatore: Dott. Andrea Tironi

Tesi di:

Alberto Bertoncini

Matricola: 908920

Anno Accademico 2019-2020

dedicato a

Aaron Bertoncini

Il tuo ricordo mi sprona a superare ogni traguardo

Indice

Introduzione	1
0.1 - Problema reale	1
0.2 - Analisi Dati	2
0.3 - Ottimizzazione	3
0.4 - Scambio interbibliotecario come problema Pickup and Delivery	5
0.5 - Obiettivi dell'elaborato	6
0.6 - Struttura dell'elaborato	7
Capitolo I: Analisi dei dati	8
1.1 - Analisi Descrittiva	8
1.1.1 - Schedulazione originale	8
1.1.2 - Elenco delle sedi	10
1.1.3 - Storico degli ordini	10
1.2 - Analisi esplorativa	13
1.2.1 - Localizzazione	13
1.2.2 - Valutazione dei costi della schedulazione	15
1.2.3 - Analisi dello storico dei prestiti	16
Capitolo II: KPI e modelli di ottimizzazione	19
2.1 - Modelli di programmazione	23
2.2 - Modello OSS	24
2.3 - Modello FOSS	27
Capitolo III: Implementazione	30
3.1 - Algoritmi di ottimizzazione	30
3.1.1 - Partizionamento	32
3.1.2 - Rimescolamento greedy	33
3.1.3 - Reinsertion (RS)	33
3.1.4 - 2-opt	34
3.1.5 - Multi Vehicle Local Search (MVLS)	35
3.2 - Descrizione tecnica	37
Capitolo IV: Analisi sperimentale	44
4.1 - Distribuzioni degli ordini	44
4.1.1 - Aggregazione degli ordini per biblioteca e per data	44
4.1.2 - Aggregazione giorno e coppia di sedi	49
4.1.3 - Aggregazione per coppia di biblioteche	52
4.1.4 - Considerazioni finali	52
4.2 - Valutazione delle soluzioni	53

4.2.1 - Soluzioni derivanti dalla schedulazione fornita	54
4.2.2 - Schedulazioni con base calcolata	59
4.2.3 - Confronto totale	65
Capitolo V: Conclusioni	68
Bibliografia - Sitografia	69
Ringraziamenti	70

Introduzione

0.1 - Problema reale

Il progetto presentato in questo elaborato è motivato dall'analisi sulla qualità dei servizi del sistema di scambio interbibliotecario del cremonese. Il termine “servizio interbibliotecario” indica la possibilità di prestito e la fornitura di documenti tra varie biblioteche. Grazie ad esso un utente può avere accesso a libri o ricevere copie di documenti posseduti da altre sedi bibliotecarie nazionali o internazionali⁽¹⁾.

Il problema in esame riguarda il loro sistema di trasporto materiale attraverso la rete bibliotecaria, in particolare come organizzare nella settimana le visite di ritiro/consegna alle varie sedi aderenti, al fine di soddisfare le richieste di prestito. L'elaborato verte sull'analisi dei dati forniti (riguardanti la posizione delle sedi, l'attuale schedulazione e lo storico degli ordini) e sull'analisi delle soluzioni. Queste ultime saranno ottenute tramite l'impiego di euristiche e modelli che puntano al miglioramento dei parametri più importanti, definiti tramite analisi dei dati.

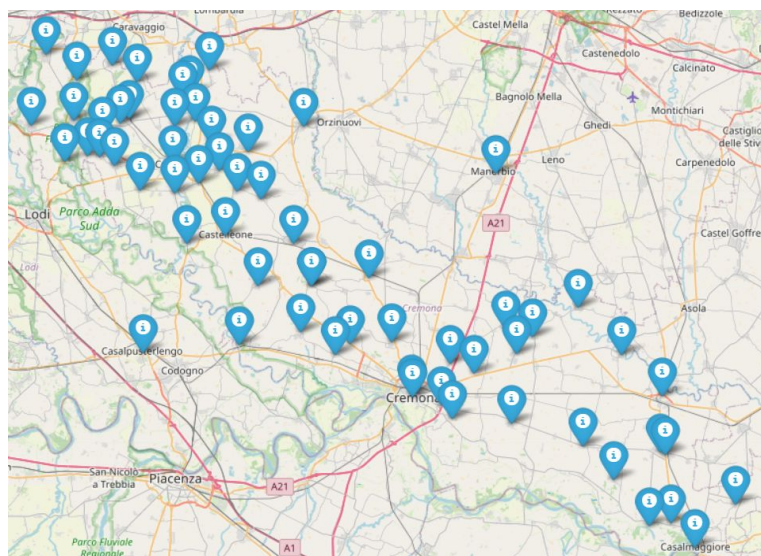


Figura 0.1.1 Disposizione delle sedi aderenti alla rete di scambio

Gli aspetti analizzati delle possibili soluzioni riguardano sia le risorse impiegate (in termini di tempo, distanza e carico di lavoro), sia il grado di servizio che la rete può fornire. Quest'ultimo non comporta direttamente spese, tuttavia migliorarlo può aumentare la soddisfazione degli utenti nei confronti della rete; con il termine “grado

di servizio” viene indicato il tempo di attesa prima che una richiesta di prestito venga soddisfatta. Lo scopo dell’elaborato sarà evidenziare un’eventuale importanza dell’analisi preliminare alla fase di miglioramento e di un’analisi delle soluzioni per la scelta della più vantaggiosa.

0.2 - Analisi Dati

L’analisi dei dati è il processo attraverso cui questi ultimi vengono ispezionati, puliti ed elaborati al fine di ottenere informazioni utili a supporto di decisioni.

Nella vita quotidiana la eseguiamo ogni qualvolta serva intraprendere una scelta, ripensando ad esperienze passate, prendendo in considerazione i vecchi risultati e le circostanze in cui la decisione è stata presa, confrontandole con le attuali e puntando sull’opzione più proficua. Risulta necessaria una metodologia più rigorosa per essere utile a decisioni di alto livello, un processo che ci permetta di estrarre le informazioni di interesse dai nostri dati, aggregandole in risultati: gli indici.

Con regole precise, modelli predefiniti ed una teoria da seguire si è in grado di effettuare un’analisi efficace nel restituire indici validi al supporto di decisioni in vari ambiti lavorativi.

L’attività di analisi svolge un ruolo fondamentale a supporto di decisioni organizzative, al fine di ottenere miglioramenti principalmente in termini di efficienza, di tempo o di costi⁽²⁾. Un’alta percentuale di aziende (98.6%) cerca di raggiungere un modello decisionale basato sull’analisi dei dati, tuttavia solo una piccola parte (32,4%) riesce ad integrarlo nella propria attività⁽³⁾, portandola ad un livello di sicurezza decisionale nettamente migliore (e più sicuro) rispetto alla concorrenza che non ne usufruisce.

L’analisi dei dati al supporto aziendale è stata diffusa negli ultimi decenni, con l’avvento di tecnologie utili a facilitarne l’uso e a costi relativamente accessibili, tuttavia la raccolta ed elaborazione dei dati avviene da millenni⁽⁴⁾ con fini logistici, sanitari e militari.

Uno dei più antichi esempi a nostra disposizione è il censimento sumerico, scritto su tavolette di argilla, il quale permise di tenere conto della cardinalità della popolazione e di avere un supporto numerico su come dividere le razioni di cibo. Un lavoro simile è stato ripreso in Egitto, dove i dati ricavati dal censimento furono utilizzati per

distribuire la forza lavoro e dai romani per tenere sotto controllo la crescita demografica nel loro territorio o per distribuire meglio le forze militari.

Un evento più recente riguarda il dottor John Snow, un fisico che nel XIX secolo aiutò a combattere la diffusione del colera, il cui contagio si era pensato avvenisse tramite aria infetta: miasma. Il dottor Snow raccolse i dati riguardanti i casi di colera e la loro posizione ed analizzandoli riuscì ad evidenziare il fatto che il problema potesse risiedere nella pompa locale dell'acqua, la quale venne bloccata e di conseguenza il numero di contagi iniziò a diminuire data l'assenza di acqua infetta in circolazione.

Durante la seconda guerra mondiale Abraham Wald, matematico ungherese, confrontò i danni degli aerei tornati salvi in base con quelli dei resti distrutti ed evidenziò il fatto che alcuni particolari punti erano sempre intatti nei velivoli funzionanti; ciò permise all'esercito americano di rinforzare queste posizioni strategiche senza appesantire troppo gli aerei, rendendoli però più sicuri.

Nella storia più recente sono stati sviluppati metodologie molto più complesse della semplice osservazione, portando l'analisi ad un livello superiore e rendendola la base della maggior parte delle decisioni economico finanziarie. Le tecnologie in sviluppo anche in questo momento permetteranno di effettuare valutazioni molto più precise e su larga scala, basti pensare alla crescita delle intelligenze artificiali che pian piano stanno sempre più automatizzando il processo di analisi, rendendo fruibile ai molti la possibilità di sfruttarne i vantaggi.

In ognuno di questi casi la prassi risulta strutturata nel seguente modo: raccolta dei dati, pulizia, elaborazione ed estrazione delle informazioni utili a livelli con metodologie più o meno sofisticate. Questo sono solo alcuni esempi di come, decisioni fondamentali per il benessere e lo sviluppo possano essere fatte coscienziosamente previa un'attenta analisi.

0.3 - Ottimizzazione

La sola analisi dei dati può dare alcuni indici utili, tuttavia l'integrazione di un modello matematico atto all'ottimizzazione può portare a risultati nettamente migliori, riuscendo non solo ad aiutare l'utilizzatore nel prendere decisioni supportate da indicatori significativi, ma anche fornendo possibili soluzioni a diversi problemi rappresentabili tramite opportuni modelli matematici.

Il termine “modello matematico” indica la descrizione di un fenomeno attraverso linguaggio formale matematico⁽⁵⁾. L'obiettivo è quindi riuscire a rappresentare un fenomeno il più precisamente possibile, in modo tale da poter esplorare con strumenti matematici il modello generato, andando a ricercare un'eventuale soluzione. In particolare, per problemi economici, logistici o organizzativi, la rappresentazione matematica viene sfruttata, più che per ricercare una soluzione, per trovare la migliore tra le soluzioni possibili; non ci basta trovare un modo di risolvere il problema, ma vogliamo la soluzione che ci dia i maggiori benefici con le minori spese. L'obiettivo di un modello può variare a seconda del sistema che si sta rappresentando e da cosa si sta cercando. La funzione obiettivo rappresenta ciò che si vuole ottenere ed in particolare se minimizzare o massimizzare una certa quantità (minimizzare i costi/tempi, massimizzare i profitti). In base all'obiettivo verranno create delle variabili per rappresentare le scelte da effettuare, come quantità da scegliere, o assegnazioni da fare. Una variabile è solitamente legata ad uno o più indici, per semplificarne l'uso, che indicheranno quanto vale quella variabile nelle circostanze indicate dagli indici e.g. la variabile x_i indica se nel giorno i viene effettuata una certa azione, oppure x_{ij} indica la quantità di materiale da trasportare da i a j . Ognuna di esse dovrà essere assegnata ad un dominio, ovvero un insieme di valori che può assumere, finito o infinito. Queste variabili sono poi utilizzate nella funzione obiettivo, in modo tale da minimizzare/massimizzare una loro espressione. La rappresentazione di un fenomeno ha però anche dei vincoli da esprimere, ovvero limitazioni imposte dal sistema, importanti da rappresentare perché senza di essi non esisterebbe il “problema”, dato che basterebbe prendere la quantità massima (a volte infinita) o minima (0 o addirittura negativa). I vincoli utilizzano variabili e costanti per limitare i valori delle prime, sia dall'alto che dal basso. L'insieme di questi elementi: funzione obiettivo, variabili, vincoli costituisce il modello del problema.

La branca della ricerca operativa che si occupa di fornire modelli e metodi risolutivi a problemi di assegnamento, allocazione e quantificazione delle risorse è la programmazione lineare. Creando modelli generici applicabili, singolarmente o combinati, ai problemi affrontati si permette la risoluzione di varie categorie di problematiche, con un semplice adattamento dei dati alle variabili utilizzate nel problema.

Uno dei problemi più affrontati è quello dei trasporti (Pickup and Delivery Problem PDP), in particolare l'ordinamento dei vari luoghi da visitare in modo tale da minimizzarne tempi o costi. Questa tipologia rappresenta tutte quelle situazioni in cui l'obiettivo è la ricerca dell'insieme ottimale di percorsi che un insieme di veicoli deve percorrere per consegnare merce ad un determinato insieme di locazioni/clienti⁽⁶⁾. A seconda delle esigenze sarà una massimizzazione o minimizzazione, nel primo caso dei profitti, nel secondo (più comune) si vorranno diminuire i costi per carburante, tempo di lavoro o distanze percorse.

Questo modello verrà preso in considerazione per una specifica istanza del problema presentatoci, il miglioramento degli scambi interbibliotecari della rete cremonese.

0.4 - Scambio interbibliotecario come problema Pickup and Delivery

Le richieste di prestito possono essere effettuate presso una delle biblioteche aderenti al servizio, che fungeranno da intermediari con la rete per inviare la richiesta alla sede che possiede il tomo ricercato. A seconda del giorno del prestito, del giorno in cui la biblioteca a cui è stato chiesto verrà raggiunta da uno dei mezzi e di quello in cui la biblioteca richiedente verrà servita con la consegna, si potrà determinare il tempo di attesa per il completamento dell'ordine.

La difficoltà maggiore non sta nel recepire gli ordini ed informare le altre biblioteche, ma nell'effettivo trasporto e consegna del materiale dato che, una buona organizzazione delle sedi può migliorare i tempi di attesa dei prestiti e diminuire i costi di trasporto.

Questo elaborato parlerà dell'analisi di dati e la loro manipolazione tramite modelli matematici.

Risulta importante essere in possesso di una base concreta, che permetta agli organi decisionali di effettuare scelte favorevoli al benessere dell'azienda, per questo tutte le informazioni raccolte o fornite devono essere trattate, prima di poter essere analizzate. Le varie biblioteche presenti in questa rete verranno raggiunte da due veicoli, che preleveranno e consegneranno i libri richiesti, secondo una precisa schedulazione delle sedi durante la settimana lavorativa.

Il problema è quindi quello di trovare un algoritmo con cui distribuire nella settimana tutte le occorrenze delle biblioteche aderenti all'iniziativa di interscambio, tenendo

conto che ogni giorno dovranno partire da un deposito iniziale (lo stesso) ed arrivare a fine giornata ognuno ad un rispettivo deposito finale.

Ogni possibile procedura dovrà prendere in input alcuni dati (fornitici dalla rete bibliotecaria del cremonese) come il numero di comparizioni di ogni biblioteca nella schedulazione ed i giorni di lavoro dei conducenti.

Verranno sfruttate anche altre informazioni (estratte dai dati forniti) ed utilizzate per cercare la migliore delle soluzioni, come le varie distanze tra le città (in tempo e chilometri) ed il peso di un collegamento (in base al numero di prestiti tra quelle due biblioteche).

La funzione obiettivo del modello utilizzato punterà a minimizzare il costo (in termini di tempo) dei vari tragitti. Le variabili riguarderanno l'appartenenza o meno di una sede in un certo giorno. I vincoli invece presenteranno limitazioni sulla numerosità delle città in un dato giorno (il che porta anche ad una limitazione numerica delle stesse) ed imporranno un certo numero di visite durante la settimana per ogni sede.

0.5 - Obiettivi dell'elaborato

- 1) Analisi del dataset reale fornito dalla rete bibliotecaria cremonese tramite analisi descrittiva ed esplorativa;
- 2) Progettazione di indicatori di performance per valutare e confrontare correttamente le soluzioni;
- 3) Creazione modelli di ottimizzazione per l'assegnamento delle sedi nella settimana lavorativa, basati sullo storico degli ordini, in modo da generare una base ottimale su cui applicare le euristiche;
- 4) Presentazione ad alto livello degli algoritmi euristici impiegati;
- 5) Analisi della distribuzione delle richieste di prestito secondo lo storico fornito;
- 6) Valutazione di potenziali soluzioni alternative a quella in uso tramite l'utilizzo dei mezzi sopra citati.

0.6 - Struttura dell'elaborato

- 1) Nel primo capitolo verrà presentata l'analisi dei dati, sia forniti che estratti;
- 2) Nel secondo capitolo verrà spiegato come costruire un indicatore delle performance e verranno analizzati i due modelli utilizzati per l'assegnamento delle sedi alle varie giornate;
- 3) Nel terzo capitolo verranno presentati gli aspetti implementativi dell'analisi quali i linguaggi, librerie, scelte progettuali e descrizione delle euristiche impiegate per ottenere i risultati da analizzare;
- 4) Nel quarto capitolo saranno riportati i risultati dell'analisi sperimentale e confrontate le varie soluzioni;
- 5) Nel quinto capitolo saranno tratte le conclusioni dai risultati degli esperimenti condotti.

Capitolo I: Analisi dei dati

Un'operazione preliminare ad ogni elaborazione statistica è l'analisi dei dati. Questo permetterà di farsi un'idea della loro distribuzione, struttura, caratteristiche e di evidenziare eventuali relazioni; inoltre possono essere individuati outlier ed errori in modo tale da “pulire” tutte le informazioni in nostro possesso, mantenendo una collezione di dati il più possibile coerente con la realtà⁽⁷⁾.

In seguito saranno trattate sia un'analisi descrittiva, sia un'analisi esplorativa.

Con l'analisi descrittiva si intende la trasformazione dei dati grezzi in una forma che li renda facili da capire, interpretare, organizzare, ordinare e manipolare al fine di generare informazioni utili⁽⁸⁾.

Una volta portata a termine, i dati puliti e organizzati potranno essere elaborati tramite l'analisi esplorativa con l'obiettivo di ispezionarli, alla ricerca di nuove informazioni e relazioni⁽⁷⁾ in modo da estrarre nuove collezioni di dati utili alla risoluzione del fenomeno in analisi.

Questo tipo di analisi va condotta con cura, selezionando quali aspetti approfondire e quali no, dato che relazioni ed informazioni inerenti potrebbero portare beneficio alla ricerca effettuata; l'estrazione di nuove collezioni non importanti per la risoluzione del fenomeno trattato porterebbero solo ad uno spreco di risorse.

1.1 - Analisi Descrittiva

I dati in nostro possesso sono stati forniti dalla rete bibliotecaria cremonese sotto forma di file:

- **Orario bibliobus.docx;**
- **Indirizzi_biblio_21-10-2019.ods;**
- **request_cr.csv.**

1.1.1 - Schedulazione originale

Orario bibliobus.docx rappresenta una tabella contenente:

- I giorni di attività dei mezzi come colonne: lunedì, martedì, mercoledì, giovedì, venerdì;

- Per ogni giorno una sequenza di sedi da visitare, rappresentate dal loro nome o città di appartenenza;
- Per ogni sede il rispettivo orario (indicativi) di arrivo del mezzo.

Questa è l'attuale schedulazione utilizzata dalla rete bibliotecaria cremonese, che dovrà essere migliorata per diminuire i tempi di attesa e di lavoro.

Quasi la totalità delle biblioteche (91 occorrenze totali nel file) verrà conteggiata per essere calendarizzata. Le due sedi escluse sono:

1. “Biblioteche a chiamata” dato che non risulterebbe fattibile creare una soluzione avendo un’occorrenza variabile a seconda delle richieste in tempo reale delle altre biblioteche;
2. “San Daniele Po” essendo una biblioteca virtuale, segnalata come da non servire.

Di conseguenza risultano 89 occorrenze calendarizzabili.

In ogni cella di questa tabella l'indicatore di una biblioteca è contrassegnato dal nome della città in cui si trova, se per quella località è presente una sola sede bibliotecaria. Viene invece specificato il nome dell'edificio nel caso in cui esistano più biblioteche aderenti alla rete bibliotecaria cremonese nella stessa città.

Crema presenta nella schedulazione:

1. Crema - Biblioteca Comunale;
2. Crema - Biblioteca Museo Civico.

Cremona presenta nella schedulazione:

1. Cremona - Centro Fumetto/Biblioteca Statale/Piccola Biblioteca;
2. Cremona - Liceo Manin;
3. Cremona - M.I.T.O.

Ogni biblioteca può presentare più occorrenze durante la settimana, sussistono un totale di 89 occorrenze con 64 sedi bibliotecarie; nello specifico 21 di queste si ripetono almeno 2 volte (18 biblioteche presentano 2 occorrenze, 2 biblioteche presentano 3 occorrenze ed una sola compare 4 volte).

Risulta importante sottolineare che in questo file non sono presenti, per ogni giorno, il deposito di partenza comune e i relativi depositi finali di ogni veicolo. Nella ricerca della soluzione andrà tenuto conto di queste necessarie aggiunte. Inoltre il file è unico nonostante si ricerchino soluzioni distinte per ogni veicolo, dunque andrà successivamente suddiviso in due schedulazioni.

1.1.2 - Elenco delle sedi

Indirizzi_biblio_21-10-2019.ods è un elenco delle biblioteche presenti sul territorio specificando, per ognuna di esse:

- Denominazione: Nome del comune - Nome della biblioteca/Tipologia;
- Comune di appartenenza;
- Sigla della provincia di appartenenza;
- Indirizzo: via, civico;
- CAP: Codice Avviamento Postale;
- Calendarizzata: campo si/no che indica se la biblioteca in questione è aderente alla rete bibliotecaria del cremonese.

In questo file abbiamo le informazioni riguardanti ogni sede bibliotecaria: delle 105 biblioteche presenti, 64 aderiscono alla rete bibliotecaria cremonese e saranno considerate nella ricerca della soluzione.

Ogni sede presente nel file di schedulazione **Orario bibliobus.docx** (filtrato delle due da non considerare) è presente in **Indirizzi_biblio_21-10-2019.ods**, questo fornisce confidenza nei confronti dei dati e permette di non dover inserire nuove occorrenze su biblioteche che non compaiono, facendo sì che i dati rimangano puri, così come ci sono stati forniti senza assunzioni introdotte esternamente.

Delle città calendarizzate: 61 si trovano nella provincia di Cremona, una in quella di Lodi (Casalpusterlengo), una in quella di Brescia (Manerbio) e una in quella di Mantova (Viadana). In particolare Manerbio e Viadana risultano i punti di scambio con le reti bibliotecarie delle relative province.

1.1.3 - Storico degli ordini

request_cr.csv contiene un elenco di prestiti dal 2016 al 2018.

Questo storico identifica un prestito con 4 campi:

- **loan_id**: un identificativo univoco del singolo prestito;
- **data_richiesta**: data e ora (YYYY-MM-DD HH:MM:SS) della richiesta di prestito presso la "biblioteca_destinazione";
- **biblioteca_destinazione**: nome della città in cui si trova la biblioteca richiedente il prestito;

- **biblioteca_partenza**: nome della città in cui si trova la biblioteca cedente il materiale.

Lo storico contiene ordini effettuati, divisi equamente nei 3 anni rappresentati:

- 54751 ordini nel 2016;
- 52775 ordini nel 2017;
- 52116 ordini nel 2018.

Va considerato che 2479 ordini risultano non validi a causa delle sedi indicate, come biblioteca di partenza o prestito, non più presenti nella schedulazione del 2019. Un'ulteriore porzione di ordini non utilizzati sono identificati dalle righe con **biblioteca_partenza** e **biblioteca_destinazione** uguali, rappresentanti prestiti ininfluenti per la nostra analisi; la cardinalità di quest'ultima categoria è di 6161 ordini. Questi due insiemi sono disgiunti: nel secondo non sono stati considerati casi in cui le due sedi partenza-destinazione sono uguali e non valide per la schedulazione del 2019. Abbiamo un totale di 8640 ordini che non saranno utilizzati nell'analisi, il 5.41% dello storico.

Delle sedi valide in questi scambi è utile visualizzarne dei grafici.

I seguenti boxplot possono mostrare efficacemente la distribuzione degli ordini, sia da e per ogni sede, evidenziando eventuali outlier.

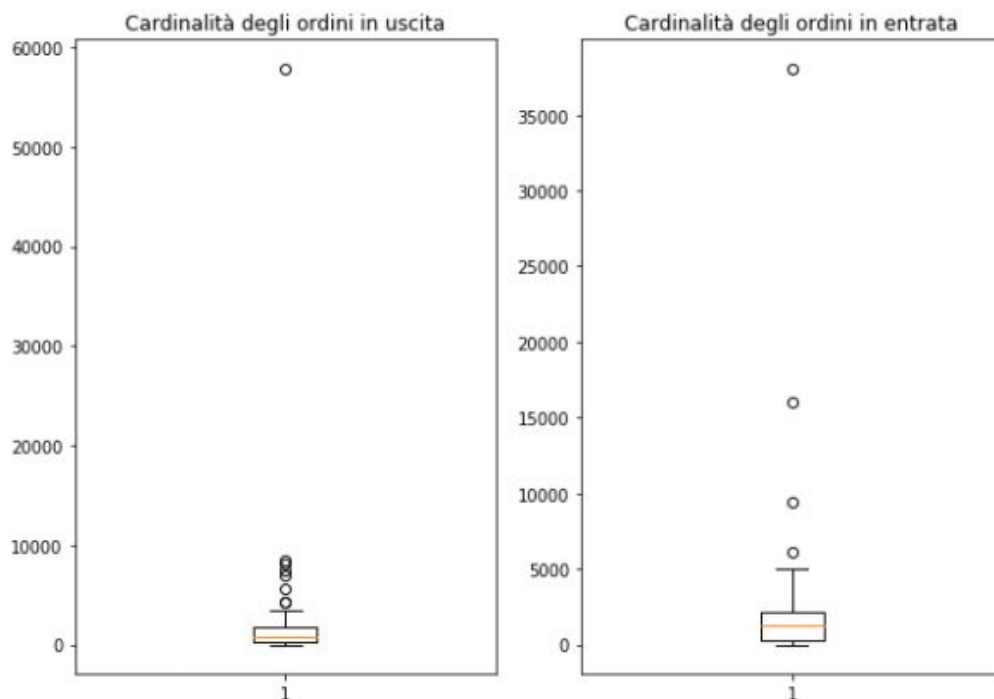


Figura 1.1.3.1 Boxplot sul numero di ordini in uscita ed entrata da ogni sede

In entrambe le rappresentazioni (figura 1.1.3.1) si notano diversi outlier, alcuni dei quali molto influenti. Nel grafico rappresentante gli ordini in uscita (biblioteche cedenti):

- Manerbio: 57912;
- Crema, Civerchi, 9: 8453;
- Rivolta d'Adda: 103;
- Castelleone: 7518;
- Offanengo: 6874;
- Casalmaggiore: 5566;
- Bagnolo Cremasco: 4328;
- Sergnano: 4265.

L'outlier più evidente è quello della biblioteca di Manerbio, con 57912 prestiti in richiesta. L'alto numero è giustificabile dal fatto che Manerbio è il punto di scambio con l'intera rete bibliotecaria di Brescia, ogni materiale derivante da questa provincia passerà da lì. Nel secondo caso abbiamo meno outlier, i cui valori sono meno marcati:

- Crema, Civerchi, 9: 38085;
- Manerbio: 16021;
- Offanengo: 9440;
- Soncino: 6111;
- Soresina: 5031;
- Pandino: 4785.

1.2 - Analisi esplorativa

Descritti i dati forniti, con eventuali correzioni o evidenziati punti critici è possibile iniziare ad estrarne ulteriori informazioni dopo aver controllato che ci sia concordanza tra le informazioni contenute nei vari file.

Delle 64 sedi (89 occorrenze) presenti in **Orario bibliobus.docx**, ora chiamato anche “Schedulazione originale”, tutte fanno riferimento ad una biblioteca presente in **Indirizzi_biblio_21-10-2019.ods**, non abbiamo ambiguità o nomi mancanti.

Ognuna delle biblioteche aderenti alla rete bibliotecaria cremonese è rappresentata almeno una volta nella schedulazione originale.

Tra le varie sedi di partenza e destinazione in **request_cr.csv**, tolte le occorrenze non collegabili ad uno degli indirizzi forniti in **Indirizzi_biblio_21-10-2019.ods**, l'unica città che non è possibile ricondurre direttamente ad una delle sedi è “Brescia”, che dovrà essere sostituita con “Manerbio, insurrezione nazionale”, il punto di scambio designato per tutti i prestiti con la provincia di Brescia e la sua rete di interscambio. Le città non riconducibili a nulla nel file degli indirizzi possono essere rimosse dall'elaborazione del file degli ordini (2016-2018) dato che le sedi in questione non sono più aderenti alla rete bibliotecaria cremonese (2019).

I file puliti non presentano incongruenze e possono essere sfruttati insieme al fine di estrarne informazioni utili.

1.2.1 - Localizzazione

Lo scopo del progetto proposto dalla rete bibliotecaria del cremonese è quello di ridurre i tempi e migliorare il servizio nella consegna dei libri, per questo bisognerà localizzare le città geograficamente al fine di poterne effettuare misurazioni come distanze temporali e spaziali.

Prendendo l'elenco delle città presenti è stato creato un nuovo file “CitiesLocation.csv” con la seguente struttura:

- **city**: Indirizzo della biblioteca in questione nel formato “NomeCittà, NomeVia/Piazza, NumeroCivico”, ricavati dal file **Indirizzi_biblio_21-10-2019.ods**;
- **lat**: la latitudine dell'indirizzo contenuto in **city**;
- **lon**: la longitudine dell'indirizzo contenuto in **city**.

Per ogni città considerata è stato ricavata la localizzazione geografica, per velocizzare la ricerca delle distanze sede-sede.

Le coordinate salvate in questo nuovo file ci permettono di localizzare su una mappa le posizioni delle nostre città e di stamparne i percorsi (figure 1.2.1.1, 1.2.1.2 e 1.2.1.3).

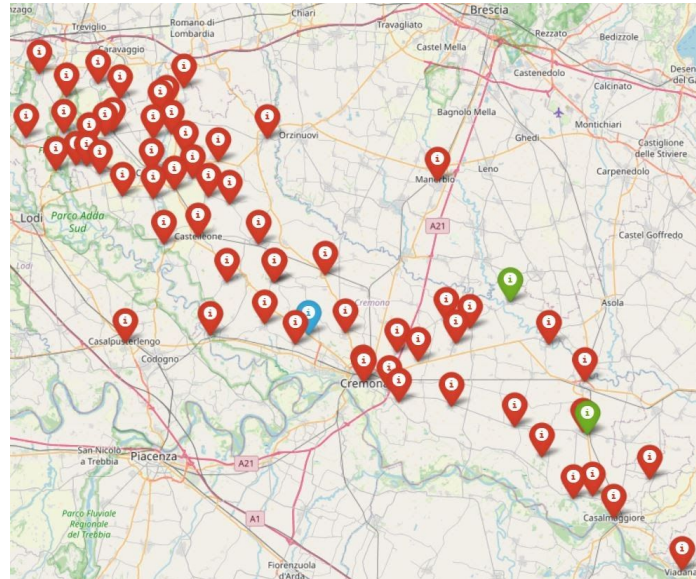


Figura 1.2.1.1 Mappa con segnaposti delle sedi bibliotecarie considerate. In blu il deposito iniziale. In verde i depositi di arrivo.



Figura 1.2.1.2 Visualizzazione dei percorsi secondo l'ordinamento fornito dalla rete bibliotecaria cremonese rispetto al primo veicolo

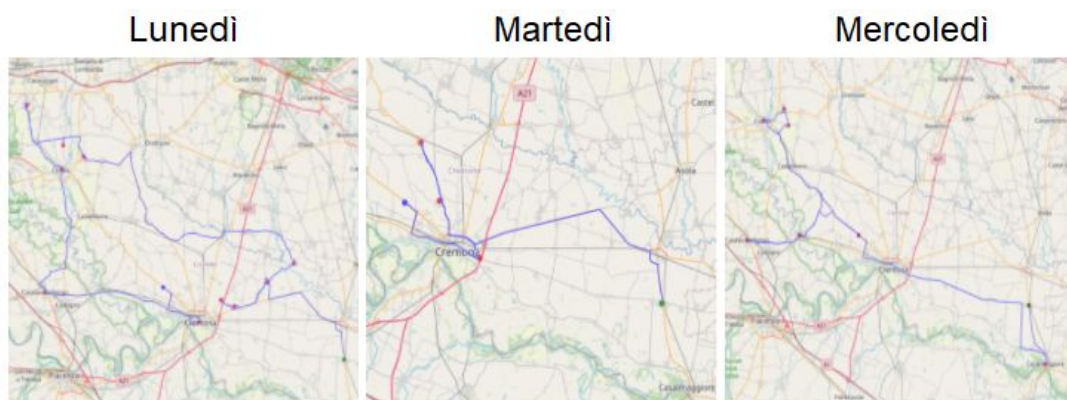


Figura 1.2.1.3 Visualizzazione dei percorsi secondo l'ordinamento fornito dalla rete bibliotecaria cremonese rispetto al secondo veicolo

La figura 1.2.1.1 mostra la posizione centrale del deposito di partenza e la disposizione delle città interessate intorno ad esso. Le sedi a nord-ovest risultano più difficili da servire essendo a maggior distanza dai depositi; al contrario, quelle intorno a Cremona risultano più facilmente integrabili nel percorso da calcolare, mantenendo costi¹ contenuti.

Il modello lavorerà sulle distanze tra le città, per questo computiamo due nuovi file:

- **corrected_times_computed.csv**: una matrice 64x64 i cui indici di colonne e righe sono le varie città. Ogni cella conterrà la distanza temporale dalla città indicata dalla riga alla città indicata dalla colonna;
- **distances_computed.csv**: una matrice 64x64 i cui come indici di colonne e righe sono le varie città. Ogni cella conterrà la distanza spaziale dalla città indicata dalla riga alla città indicata dalla colonna.

Gli indici in entrambi i file saranno costituiti dai nomi nel formato “città, via, civico” indicanti la sede della biblioteca in questione. In tutte le successive soluzioni generate i nomi rispetteranno questo formato per rendere possibile l'estrazione delle distanze dalle due matrici.

1.2.2 - Valutazione dei costi della schedulazione

Ottenute queste informazioni è possibile iniziare ad analizzare meglio la qualità della schedulazione presentata dalla rete bibliotecaria cremonese.

Al fine di semplificarne l'utilizzo, la schedulazione originale è stata divisa in due file: **route.csv** contenente la sola calendarizzazione dei nomi e **route_time.csv** contenente gli orari indicativi di arrivo dei mezzi.

¹ In termini di tempo e spazio

In primo luogo la schedulazione iniziale andrà suddivisa tra i due veicoli a disposizione (file `furgoncino1_routing.csv` e `furgoncino2_routing.csv`²) tramite euristica, dopodiché si potrà iniziare a valutare le tempistiche di questo ordinamento.

Il primo file presenta un tempo di percorrenza pari a 1005 minuti e 922 chilometri, il secondo 427 minuti di percorrenza e 382 chilometri. Questi saranno i principali parametri da migliorare grazie all'elaborazione di nuovi ordinamenti.

1.2.3 - Analisi dello storico dei prestiti

Un ulteriore parametro che può essere preso in considerazione è quello rappresentante i tempi di attesa che intercorrono tra la visita di due città arbitrariamente selezionate. Questo ci potrebbe fornire un'idea della bontà della distribuzione delle sedi nell'ordinamento preso in esame: una schedulazione con tempi di percorrenza bassi e poca strada percorsa potrebbe presentare problemi se non si tiene conto dell'indice di attesa mentre si calcola la soluzione; e.g. una soluzione con più occorrenze inserite vicine, magari lo stesso giorno diminuirebbe l'uso di risorse, tuttavia non migliorerebbe i tempi di attesa, dato che un prestito richiesto il giorno successivo alla loro visita dovrebbe attendere una settimana per essere ritirato. Una distribuzione più equilibrata dovrebbe tenere conto del tempo di attesa per migliorare l'efficienza del servizio, senza andare troppo a discapito di tempo e spazio.

Questo indice può essere calcolato partendo da una qualsiasi schedulazione, effettuando una media della distanza tra le varie combinazioni delle sedi presenti al fine di ottenere, per ogni coppia interessata, un indicatore riguardante il tempo medio di attesa per il trasporto da una all'altra. La relazione non è bivalente, si potrebbero ottenere valori differenti per la coppia "SedeA"- "SedeB" rispetto alla coppia "SedeB"- "SedeA".

Un indicatore così costruito interesserebbe la disposizione delle sedi sull'arco temporale interessato, tuttavia sarebbe utile solo nel caso in cui i prestiti venissero richiesti in egual numero alle varie biblioteche. Dunque bisogna anche pesare il tempo di attesa con la numerosità di ordini effettuati da un "SedeA" ad una "SedeB", in due specifici giorni.

Abbiamo informazioni riguardanti lo storico degli ordini su un arco temporale di due anni, da questi è possibile estrarre il numero di ordini fatti da ogni biblioteca verso

² nello stesso formato di `route.csv`

un'altra un dato giorno, in modo tale da pesare diversamente i tempi di attesa per la consegna del materiale; e.g. due sedi potrebbero risultare distanti molti giorni (nella schedulazione), tuttavia se non sussistono scambi tra le due la loro lontananza non influirà sul grado del servizio.

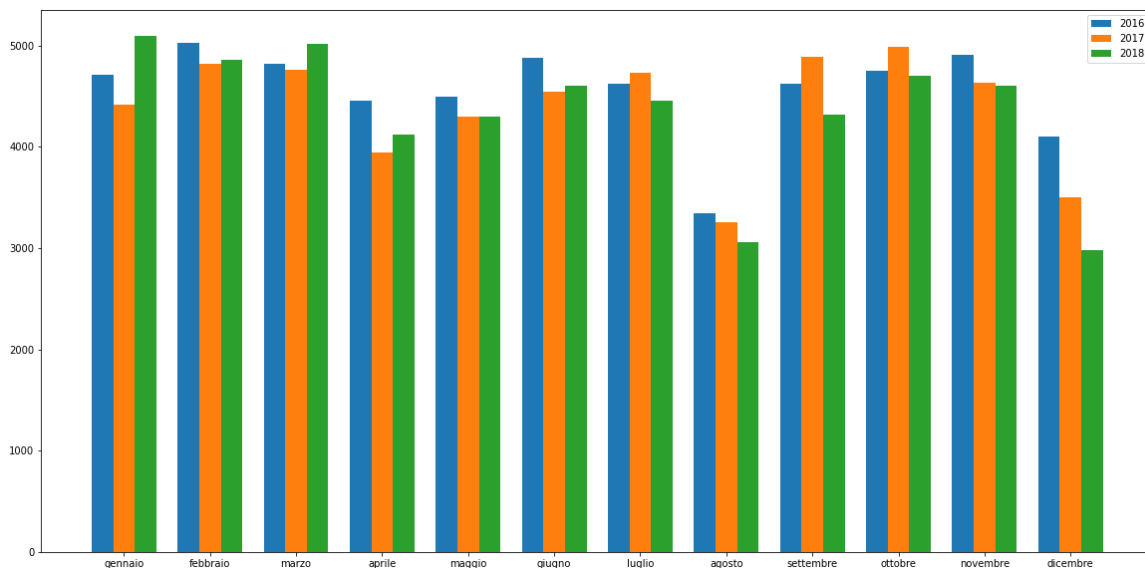


Figura 1.2.3.1 Suddivisione degli ordini di ogni anno nei vari mesi

Con l'eccezione di dicembre la divisione degli ordini nei vari mesi risulta costante negli anni, rendendo questo storico valido per estrarre le tendenze delle richieste di prestiti (figura 1.2.3.1). Per poter ricavare facilmente queste informazioni è utile elaborare i dati nel file **request_cr.csv** estraendo le occorrenze dei vari ordini e dividendoli a seconda della giornata della richiesta; otterremo un nuovo file che per ogni giorno della settimana (sabato e domenica compresi) presenterà una lista di ordini nel formato “**biblioteca_partenza-biblioteca_arrivo**”, in questo modo sarà facile ottenere la frequenza degli ordini effettuati in un dato giorno, o ad una granularità maggiore, la frequenza di ordini da una certa sede ad un'altra un dato giorno. Queste frequenze saranno salvate con il formato descritto nel file **num_orders.csv**.

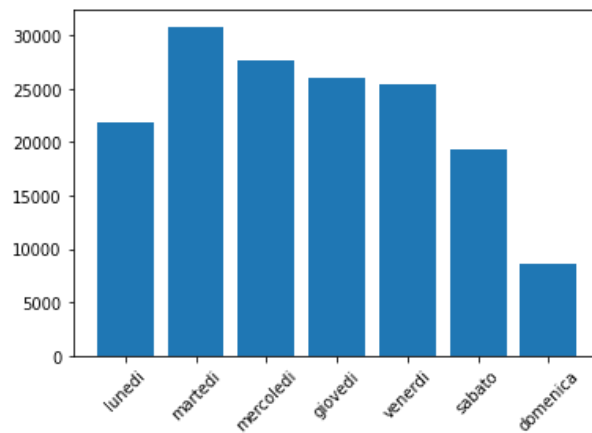


Figura 1.2.3.2 Numero di ordini divisi per giorno

La diminuzione degli ordini effettuati il sabato e (soprattutto) la domenica potrebbe essere data dal fatto che alcune sedi risultano chiuse in quei giorni (figura 1.2.3.2). Non avendo informazioni a riguardo, questi dati verranno trattati in egual modo per il calcolo di indicatori su di essi basati.

Capitolo II: KPI e modelli di ottimizzazione

Le soluzioni generate tramite algoritmi euristici³ restituiscono schedulazioni nello stesso formato di **route.csv**. Calcolati questi ordinamenti è possibile analizzarli per controllarne l'ammissibilità ed un eventuale miglioramento dei parametri in confronto alla soluzione originale in utilizzo.

I parametri che andremo ad analizzare per le varie schedulazioni generate saranno:

- Tempo di percorrenza totale;
- Distanza totale percorsa;
- Bilanciamento del lavoro tra i due veicoli in uso;
- Tempi medio di attesa prima della presa in carico di un ordine;
- Tempo medio di attesa dall'attuazione di un prestito alla relativa consegna.

Sono stati precedentemente generati i file contenenti le distanze di ogni tratto, inteso come collegamento da una sede ad un'altra, il cui valore numerico (temporale o spaziale) indica il costo di quel percorso. Risulta semplice calcolare i primi due indici, basterà infatti, per ogni schedulazione, accedere alle rispettive matrici delle distanze per poi sommare i pesi dei vari tratti.

Definiamo le matrici temporale e spaziale rispettivamente: T_{ij} e D_{ij} in cui i e j sono i nomi di due sedi utilizzate come indici per accedere alla distanza dalla sede i alla sede j della matrice utilizzata.

Le giornate lavorative (lunedì,..., venerdì) possono essere riscritte nell'insieme ordinato $G=\{0,...,4\}$.

I veicoli verranno rappresentati dall'insieme $V=\{1,2\}$.

Definiamo $O_{g,v}$ come l'ordinamento di un dato giorno $g \in G$ e veicolo $v \in V$ e $O_{g,v,i}$ come l'accesso alla sede in posizione i nel giorno $g \in G$ per il veicolo $v \in V$.

Per il veicolo $v \in V$ che non effettua lavori nel giorno $g \in G$ $O_{g,v}$ risulterà un insieme vuoto.

³ Trattati nel capitolo 3

Si possono calcolare il tempo e lo spazio impiegati un dato giorno $g \in G$ con le seguenti definizioni:

$$\text{Tempo_g}(\mathbf{O}, \mathbf{v}, \mathbf{g}): \sum_{i=0}^{|O_{g,v}|-1} T_{O_{g,v,i}} + T_{O_{g,v,i+1}}$$

$$\text{Spazio_g}(\mathbf{O}, \mathbf{v}, \mathbf{g}): \sum_{i=0}^{|O_{g,v}|-1} D_{O_{g,v,i}} + D_{O_{g,v,i+1}}$$

Dato un veicolo $v \in V$ si calcolano i suoi parametri temporale e spaziale come:

$$\text{Tempo_v}(\mathbf{O}, \mathbf{v}): \sum_{g \in G} \sum_{i=0}^{|O_{g,v}|-1} T_{O_{g,v,i}} + T_{O_{g,v,i+1}}$$

$$\text{Spazio_v}(\mathbf{O}, \mathbf{v}): \sum_{g \in G} \sum_{i=0}^{|O_{g,v}|-1} D_{O_{g,v,i}} + D_{O_{g,v,i+1}}$$

Volendo invece ottenere il tempo o la distanza totale su una settimana di lavoro:

$$\text{Tempo_tot}(\mathbf{O}): \sum_{g \in G} \sum_{v \in V} \sum_{i=0}^{|O_{g,v}|-1} T_{O_{g,v,i}} + T_{O_{g,v,i+1}}$$

$$\text{Spazio_tot}(\mathbf{O}): \sum_{g \in G} \sum_{v \in V} \sum_{i=0}^{|O_{g,v}|-1} D_{O_{g,v,i}} + D_{O_{g,v,i+1}}$$

Dall'indice temporale è possibile verificare l'ammissibilità di un ordinamento, andando a controllare se il tempo di percorrenza sfiora l'orario lavorativo dei veicoli oltre che per estrarre le tempistiche di lavoro.

Nel caso del bilanciamento si calcolerà la differenza del lavoro effettuato (in termini di tempo o spazio) tra i due veicoli, dividendo il risultato per il lavoro totale. Questo indicatore varrà solo per i giorni in cui lavorano entrambi i veicoli, andando a evidenziare eventuali sovraccarichi su uno dei due mezzi. Più il valore verrà minimizzato/piccolo meglio sarà bilanciato il lavoro in una determinata giornata lavorativa.

Scelto un ordinamento \mathbf{O} e su quale matrice \mathbf{M} (tempo o distanza) calcola l'indice:

$$\text{Bilanciamento}(\mathbf{O}, \mathbf{M}): \frac{\text{abs}((\sum_{g \in G \setminus \{0,4\}} \sum_{i=0}^{|O_{g,1}|-1} M_{O_{g,v,i}} O_{g,v,i+1}) - (\sum_{g \in G \setminus \{0,4\}} \sum_{i=0}^{|O_{g,2}|-1} M_{O_{g,v,i}} O_{g,v,i+1}))}{\sum_{g \in G \setminus \{0,4\}} \sum_{v \in V} \sum_{i=0}^{|O_{g,v}|-1} M_{O_{g,v,i}} O_{g,v,i+1}}$$

Gli ultimi due indici vengono riassunti col termine “*grado di servizio*”, ovvero gli aspetti che più toccano i clienti delle biblioteche.

Il tempo medio per la presa in carico di un ordine e per il suo completamento sono simili nella procedura di calcolo: il primo sarà il risultato di una media tra tutti i tempi di attesa che intercorrono da ogni ordine presente in **num_orders.csv**⁴ alla prima presa in carico del materiale da parte della rete bibliotecaria; definiamo il tempo di attesa tra la sede “A” e la sede “B” come il numero di giorni che intercorrono tra la giornata in cui avviene la richiesta di prestito e la prima occorrenza di “B”, iniziando a contare dal giorno successivo alla giornata di “A”. Questa distanza, in giorni, tiene conto di sabato e domenica qualora la prima occorrenza della sede cedente si trovasse la settimana successiva⁵.

Avendo uno storico degli ordini **S** contenenti gli ordini nel formato (sede_cedente, sede_richiedente, giorno_prestito).

Definita la funzione per la distanza in giorni tra **g** e **k** come:

$$\text{dist}(g,k)$$

La funzione per ottenere, dato un giorno **k**, il primo giorno utile per la raccolta del materiale nell'ordinamento **O** per la sede **i** come:

$$\text{raccolta}(\mathbf{O}, k, i)$$

Il tempo medio per la presa in carico del materiale, dato uno storico **S** ed un ordinamento **O** verrà calcolato come:

$$\text{media_raccolta: } \frac{\sum_{(r,c,g) \in S} \text{dist}(g, \text{raccolta}(\mathbf{O}, g, r))}{|S|}$$

L'indice del tempo per la presa in carico può evidenziare una scorretta traslazione dell'ordinamento settimanale, che magari risulta ottimale per i tempi di presa in carico e consegna, tuttavia presenta rallentamenti nei tempi che intercorrono tra la richiesta di prestito e la presa in carico. Supponiamo di avere la soluzione ottimale in termini di tempi di attesa tra il ritiro del materiale in una sede “A” e la consegna in “B”. Se questa soluzione non tiene conto anche del momento in cui i prestiti vengono richiesti alle biblioteche, il tempo intercorso tra l'effettuazione dell'ordine e l'inizio del trasporto

⁴ Questo per semplificare l'estrazione del giorno della settimana in cui è effettuata la richiesta del prestito, informazione salvata in questo file

⁵ Si ha una visione ciclica delle settimane, alla fine di una inizia un'altra uguale.

materiale potrebbe incidere sul grado di servizio. Più nel dettaglio, se due sedi “A” e “B” si trovassero ad un giorno di distanza (giovedì e venerdì rispettivamente), con richieste da “A” a “B” effettuate prevalentemente il lunedì, il tempo totale di attesa per il completamento dell’ordine aumenterebbe.

L’ultimo indice rappresenta la media dei tempi di servizio, dalla domanda di prestito alla consegna presso la biblioteca richiedente. Per ogni ordine si calcola il tempo d’attesa dal giorno dell’ordine a quello di consegna, sarà poi sufficiente effettuare una media di queste distanze. Il primo giorno disponibile per la consegna è il successivo alla visita della sede cedente il materiale in cui compare un’occorrenza della sede richiedente.

Mantenendo le definizioni utilizzate precedentemente, basta aggiungere una funzione per ottenere il primo giorno utile di consegna dato un ordinamento O , un giorno da cui partire a contare k ed una sede interessata alla consegna i :

La funzione per ottenere il primo giorno utile per la consegna del materiale in una sede i , dato un giorno k di partenza e un ordinamento O :

consegna(O, k, i)

Il tempo medio per il completamento di un ordine, dato uno storico S ed un ordinamento O verrà calcolato come:

$$\frac{\sum_{(r,c,g) \in S} dist(g, consegna(O,c,raccolta(O,g,r)))}{|S|}$$

Tutte le analisi e gli indici ricavati da esse potranno essere usati per valutare le soluzioni ottenute tramite vari algoritmi euristici⁶, tuttavia la loro integrazione in modelli di ottimizzazione può portare ad una soluzione di base ottimale per poter estrarre schedulazioni migliori.

Questi indici costituiranno i nostri “Key Performance Indicator” (KPI), ovvero indicatori di prestazione. Un KPI è generalmente un valore numerico, in una scala o meno, che permette di valutare la bontà di un determinato aspetto del processo in esame.

Nel caso trattato andremo ad esaminare aspetti riguardanti le risorse utilizzate (tempo e spazio), il buon utilizzo dei veicoli a disposizione e l’ottimalità della disposizione delle sedi nella settimana in relazione con lo storico degli ordini.

⁶ Trattati nel capitolo 3

2.1 - Modelli di programmazione

Mentre lo sfruttamento delle risorse verrà ottimizzato tramite algoritmi euristici, l'ottimizzazione della distribuzione delle sedi nella settimana può essere ricavata integrando una fase di modellizzazione sul problema della distribuzione; i risultati forniranno una base ottimale per le euristiche.

Di seguito è riportata una descrizione formale delle parti di un modello di programmazione⁽⁹⁾:

- **Parametri:** sono i dati del problema e rappresentano delle quantità fissate che dipendono dai diversi elementi del sistema;
- **Variabili:** sono le grandezze del sistema di cui non conosciamo il valore e sulle quali possiamo agire per determinare diverse soluzioni alternative del problema;
- **Funzione obiettivo:** è la quantità da massimizzare o minimizzare, descritta tramite combinazioni di variabili e parametri;
- **Vincoli:** sono delle relazioni che descrivono le condizioni di ammissibilità delle soluzioni. Servono quindi per discriminare le combinazioni di valori delle variabili che generano soluzioni accettabili del problema, da quelle che non lo sono.

Il problema che andremo a modellare riguarda l'assegnamento delle sedi nei giorni per migliorare il grado di servizio, ovvero abbattere i tempi di attesa per la conclusione di un prestito. Nello specifico verranno creati due modelli: il primo in cui si cercherà di ottimizzare il tempo di attesa per la consegna pesandolo col numero di prestiti tra le sedi ricavato dallo storico degli ordini; il secondo in cui si stimerà più finemente il peso della distanza tra due sedi, prendendo in considerazione solo le richieste effettuate alla biblioteca cedente nei giorni intercorsi tra le giornate di cui si cerca la distanza, non il numero complessivo.

2.2 - Modello OSS

Optimized Start Schedule, così verrà chiamato il primo modello e presenterà la seguente struttura:

Parametri:

- ◊ **B** insieme delle sedi bibliotecarie;
- ◊ **V_i** indica il numero di occorrenze della città **i** ∈ **B** nella schedulazione;
- ◊ **G** insieme dei giorni da lunedì a venerdì;
- ◊ **C_g** indica la capacità del giorno **g** ∈ **G** in termini di sedi inserite;
- ◊ **F_{i,g,j,k}** indicherà la distanza pesata sul numero di ordini dalla biblioteca **i** ∈ **B** nel giorno **g** ∈ **G** alla biblioteca **j** ∈ **B** nel giorno **k** ∈ **G**.

Variabili:

- ◊ **x_{i,g}** ∈ {0,1} indicherà se la biblioteca **i** ∈ **B** apparirà nel giorno **g** ∈ **G**;
- ◊ **y_{i,g,j,k}** ∈ {0,1} indicherà se la distanza pesata da **i** ∈ **B** in **g** ∈ **G** a **j** ∈ **B** in **k** ∈ **G** andrà considerata o meno.

Funzione obiettivo:

$$[1] \quad \min \sum_{i \in B} \sum_{g \in G} \sum_{j \in B, j \neq i} \sum_{k \in G} F_{i,g,j,k} y_{i,g,j,k} .$$

Vincoli:

$$[2] \quad \sum_{g \in G} x_{i,g} = V_i \quad \forall i \in B ;$$

$$[3] \quad \sum_{i \in B} x_{i,g} \leq C_g \quad \forall g \in G ;$$

$$[4] \quad x_{i,g} + x_{j,k} - 1 \leq y_{i,g,j,k} \quad \forall i \in B, \quad \forall g \in G, \quad \forall j \in B, \quad \forall k \in G .$$

Tra i parametri forniti in input al modello si evidenziano **C_g** e **F_{i,g,j,k}**.

C_g indica il numero massimo di città assegnabili ad un giorno **g**, questo per evitare una concentrazione troppo elevata di sedi un dato giorno, che porterebbe la soluzione ad essere poco equa nella distribuzione delle biblioteche durante la settimana. I giorni martedì, mercoledì e giovedì risulteranno avere una capacità maggiore a seguito della

presenza del doppio dei veicoli rispetto a lunedì e martedì, che presentano un solo mezzo di trasporto.

I valori di $F_{i,g,j,k}$ sono calcolati come la distanza in giorni da g a k moltiplicata per il numero di prestiti dalla biblioteca i alla j . Più nel dettaglio:

Definiamo $\text{dist}(g, k)$ come una funzione che presi due giorni della settimana in input restituisce la sequenza di giorni da g (compreso) a k non compreso. Nel caso in cui k venga prima di g (g ="venerdì", k ="lunedì") verrà restituita la sequenza da g a fine della settimana ("domenica") aggiungendo poi i giorni mancanti per arrivare a k .

Definiamo $\text{tot_ord}(i, j)$ come una funzione che prende in input le due biblioteche interessate e restituisce il numero di prestiti da i a j presenti nello storico.

$$F_{i,g,j,k} = \text{dist}(g, k) * \text{tot_ord}(i, j)$$

La funzione obiettivo [1] cercherà di minimizzare la somma delle distanze pesate tra le biblioteche nei vari giorni, solamente per le quadruple (i,g,j,k) presenti nella schedulazione con i, j sedi bibliotecarie diverse tra loro e g, k giorni della settimana lavorativa. Per i collegamenti non presenti nell'ordinamento generato, la variabile $y_{i,g,j,k}$ ha valore nullo e quindi porta all'annullamento del contributo della corrispettiva $F_{i,g,j,k}$.

Il vincolo [2] forza il modello ad assegnare per ogni biblioteca i esattamente V_i occorrenze, sommando tutti i valori di $x_{i,g}$ nei vari giorni. Essendo il $x_{i,g}$ 1 solamente se la biblioteca i è presente il giorno g sommando attraverso i giorni otterremo il numero di occorrenze assegnate dal modello alla sede i . Non vengono contemplate visite multiple alla stessa sede nello stesso giorno.

Il vincolo [3] invece rende inammissibili tutte le soluzioni che sfiorano le capacità assegnate ai vari giorni. Sommando $x_{i,g}$ sulle biblioteche, dato un giorno g ci si assicura che la loro cardinalità sia minore della capacità massima per quel giorno.

Il vincolo [4] lega le variabili in modo tale che non risultino in contraddizione: i valori di x limitano dal basso quelli delle y , o viceversa le y limitano dall'alto i valori delle x .

Il vincolo esprime la seguente relazione: se c'è un'occorrenza della biblioteca i nel giorno g ($x_{i,g} = 1$) e un'occorrenza della biblioteca j nel giorno k ($x_{j,k} = 1$) allora la distanza pesata dalla sede i in g alla sede j in k va tenuta da conto ($y_{i,g,j,k} = 1$). Questo forzerà

alcune y a valere 1, dato che senza questo vincolo potrebbero valere tutte 0, minimizzando così la somma.

L'output di questo modello saranno i valori delle x , che permetteranno di creare una schedulazione⁷ inserendo le sedi nei vari giorni.

Il modello restituisce, in base ai valori delle x , una singola schedulazione che andrà successivamente partizionata per i giorni in cui operano due veicoli. Inoltre, c'è un limite al numero massimo di sedi inseribili in una giornata, questo non ci assicura l'ammissibilità dal punto di vista dell'orario lavorativo dei due veicoli; la soluzione potrebbe risultare inammissibile, tuttavia bisogna sottolineare che il modello creato si occupa solamente dell'assegnamento giorno per giorno delle biblioteche, non di trovare il loro ordinamento ottimale nella giornata, questo compito è demandato alle euristiche che verranno applicate, partendo dal partizionamento all'ottimizzazione della soluzione.

⁷ Nello stesso formato di **route.csv**

2.3 - Modello FOSS

Notazione:

- A sede cedente il materiale;
- B sede richiedente il materiale;
- G ultimo giorno in cui è stata visitata la sede A prima di K ;
- K giorno in cui viene prelevato il materiale da A;
- D giorno in cui viene consegnato a B;

Il precedente modello non permette di pesare correttamente la distanza tra due giorni: pesando la distanza tra due giornate (K e D) con la frequenza degli ordini tra le due sedi in questione (materiale da A a B) si dà la stessa importanza ad ogni assegnamento che vede A precedente a B. Una misurazione più corretta sarebbe contare solamente la frequenza degli ordini presentati da A a B nelle giornate tra G e K; in questo modo si raggruppano tutti gli ordini da A a B presentati nei giorni tra G e K, così con un solo ritiro in A viene raccolto molto più materiale che considerando i soli ordini effettuati nel giorno G. Nel caso in cui siano presenti degli ordini nei giorni esterni all'intervallo temporale G-K vengono esclusi dal conteggio: non essendo ritirabili il giorno K non contribuiscono al peso della distanza K-D.

In figura 2.3.1 viene presentato un esempio di calcolo della distanza pesata tra giovedì e venerdì, rispetto a due sedi A e B supponendo che l'ultima occorrenza di A, prima del giorno di ritiro (giovedì), sia martedì. Essendo la sede A raggiunta il giovedì e la sede B il venerdì la distanza temporale risulterà di 1 giorno; il peso verrà calcolato come la somma dei prestiti da A a B sui giorni tra martedì e giovedì (escluso).

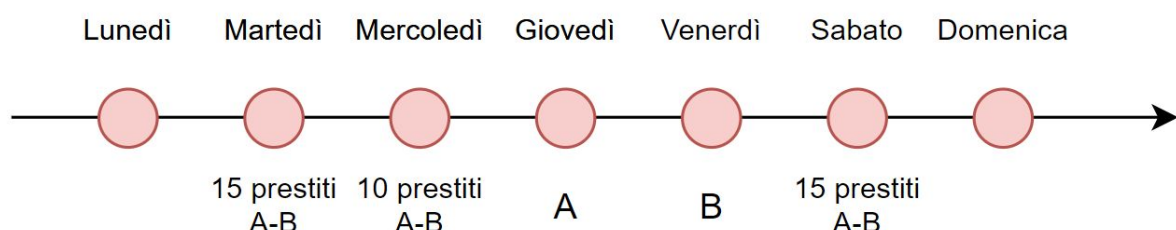


Figura 2.3.1 istanza di esempio per la stima del peso

La distanza viene quindi pesata con 25, non con la totalità di 40 prestiti.

Questa ottimizzazione è implementata nel seguente modello.

Fine Optimized Start Schedule (FOSS), questo modello utilizza una diversa funzione per stimare il peso della distanza tra due giorni, selezionate due biblioteche.

Il modello è simile al precedente:

Parametri:

- ◊ **B** insieme delle sedi bibliotecarie;
- ◊ **V_i** indica il numero di occorrenze della città **i** ∈ **B** nella schedulazione;
- ◊ **G** insieme dei giorni da lunedì a venerdì
- ◊ **C_g** indica la capacità del giorno **g** in termini di sedi inserite
- ◊ **F_{i,g,j,k}** indicherà la distanza pesata sul numero di ordini dalla biblioteca **i** ∈ **B** in tutti i giorni intercorsi tra il giorno **g** ∈ **G** alla biblioteca **j** ∈ **B** nel giorno **k** ∈ **G**.

Variabili:

- ◊ **x_{i,g}** ∈ {0,1} indicherà se la biblioteca **i** ∈ **B** apparirà nel giorno **g** ∈ **G**;
- ◊ **y_{i,g,j,k}** ∈ {0,1} indicherà se la distanza pesata da **i** ∈ **B** in **g** ∈ **G** a **j** ∈ **B** in **k** ∈ **G** andrà considerata o meno.

Funzione obiettivo:

$$[1] \quad \min \sum_{i \in B} \sum_{g \in G} \sum_{j \in B, j \neq i} \sum_{k \in G} (F_{i,g,j,k} / (v_i * v_j)) * y_{i,g,j,k}.$$

Vincoli:

$$[2] \quad \sum_{g \in G} x_{i,g} = V_i \quad \forall i \in B ;$$

$$[3] \quad \sum_{i \in B} x_{i,g} \leq C_g \quad \forall g \in G ;$$

$$[4] \quad x_{i,g} + x_{j,k} - 1 \leq y_{i,g,j,k} \quad \forall i \in B, \quad \forall g \in G, \quad \forall j \in B, \quad \forall k \in G .$$

Il modello rimane invariato nei vincoli [2], [3] e [4], cambia solo nella definizione di **F_{i,g,j,k}** e nella funzione obiettivo [1].

Questo nuovo **F_{i,g,j,k}** stima meglio il peso della distanza tra **g** e **k** valutando quanti ordini è possibile cumulare con l'assegnamento delle sedi **i** e **j** alle due giornate in questione.

Questo peso non prenderà in causa il numero totale di scambi da i a j , ma solamente quelli effettuati nei giorni tra l'ultima occorrenza di i prima di g (definita t) e g ; in particolare $F_{i,g,j,k}$ per ogni giorno nell'intervallo $[t, g)$, verrà sommato il numero di prestiti richiesti da i a j quel dato giorno.

Definiamo $\text{last_occ}(i, g)$ come una funzione che presa in input una sede i restituisce l'ultima giornata in cui è presente prima del giorno g .

Definiamo $\text{n_ord}(i, j, g)$ come una funzione che preso in input: i la sede cedente, j la sede richiedente e g la giornata in cui è effettuata la richiesta di prestito restituisce il numero di ordini da i a j il giorno g .

Manteniamo la definizione di $\text{dist}(g, k)$ del modello OSS.

$$F_{i,g,j,k} = \sum_{t \in \text{dist}(\text{last_occ}(i, g), g)} (\text{n_ord}(i, j, t) \text{dist}(t, k))$$

La funzione obiettivo [1] utilizza questo indice e lo moltiplica per la relativa $y_{i,g,j,k}$ tuttavia prima lo divide per il prodotto $v_i * v_j$, questo per distribuire il valore se esistono più combinazioni possibili tra le sedi i e j date dal loro numero di occorrenze.

Il resto dei parametri e i vincoli sono analoghi a quelli utilizzati nel modello OSS.

Capitolo III: Implementazione

3.1 - Algoritmi di ottimizzazione

Nel capitolo precedente è stato mostrato che i modelli possono distribuire, in maniera più o meno fine, le città nelle varie giornate lavorative ottimizzando il grado del servizio. Questo non assicura però ciò che inizialmente si andava ricercando: la diminuzione dei tempi e del chilometraggio.

Esistono due strade percorribili per la ricerca di una soluzione ottima:



Figura 3.1.1 Schema a moduli basato sulla schedulazione fornita

Nella prima evitiamo l'ottimizzazione del servizio, basando il grosso del lavoro sulla ricerca del riordinamento migliore delle città al fine di diminuire l'utilizzo delle risorse.



Figura 3.1.2 Schema a moduli con ottimizzazione dell'assegnamento

Con la seconda vengono rimescolate le carte, non partendo da un assegnamento occorrenza-giornata dato, ma stimandolo nel modo migliore possibile sulla base delle informazioni fornite.

La fase di raccolta dati e requisiti comprende tutte le operazioni che portano alla conoscenza del problema trattato e dei dati utili alla sua risoluzione, insieme ai vincoli da rispettare.

La fase aggiuntiva del secondo schema sfrutta i modelli trattati nel precedente capitolo per fornire un punto di partenza ottimale (schedulazione col miglior grado di servizio) su cui basare le fasi successive.

La fase di ottimizzazione si avvale degli algoritmi euristici per migliorare a vari gradi la schedulazione sottoposta in input a livello di tempo e spazio.

Infine saranno valutati tempi, distanze e grado di servizio (KPI) della schedulazione restituita.

In questa sezione saranno trattati ad alto livello gli algoritmi euristici utilizzati; inoltre sarà spiegato il loro funzionamento per la generazione delle varie soluzioni che verranno prese in esame.

Con algoritmo euristico (euristica) si intende un particolare tipo di algoritmo atto ad una risoluzione più veloce del problema in esame rispetto a metodi classici. Questi ultimi potrebbero portare a tempi di attesa eccessivi per ottenere il risultato ottimale. Vengono sfruttati anche nel caso in cui non esista un metodo esatto di risoluzione, ma si conoscono alcuni passaggi che possono portare ad un risultato approssimato all'ottimo⁽¹⁰⁾.

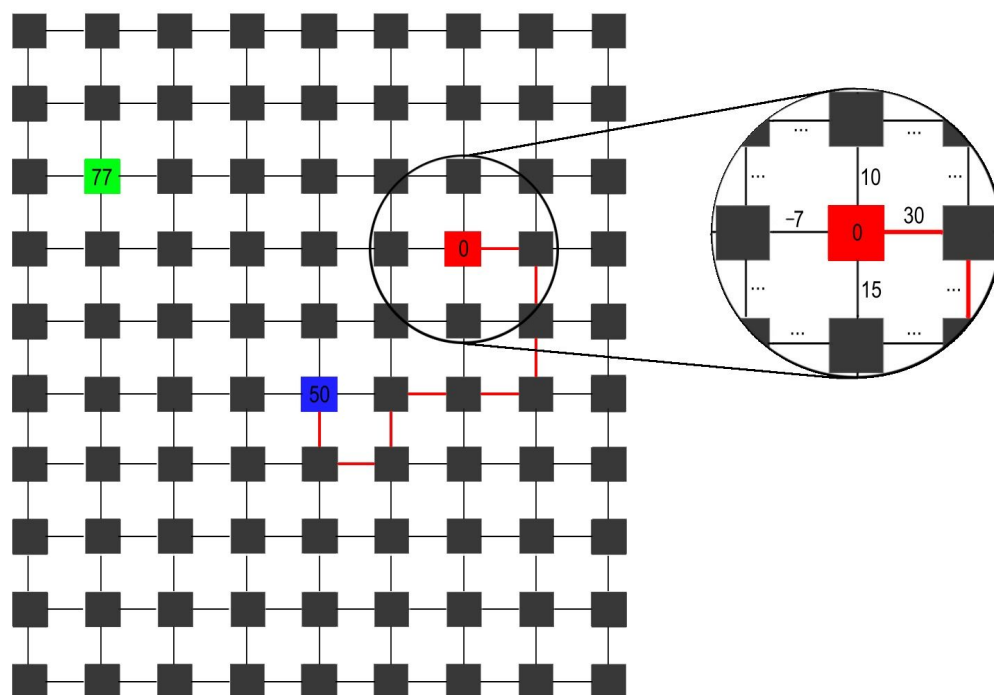


Figura 3.1.3 Concetto grafico di ricerca locale: in rosso il punto iniziale ed il percorso delle scelte effettuate, in blu la soluzione ottima (locale) trovata dalla ricerca locale e in verde l'ottimo globale.

Un algoritmo di Ricerca Locale (Local Search) è un particolare tipo di algoritmo euristico che può essere usato su problemi in cui l'obiettivo è massimizzare/minimizzare un particolare aspetto del sistema in esame, andando ad operare scelte tra un numero ristretto di candidati, appartenenti all'insieme totale delle possibilità. L'algoritmo si muove nello spazio delle soluzioni applicando modifiche

locali finché apportano migliorie o finché un dato tempo di esecuzione si è esaurito. Questo potrebbe precludere il raggiungimento dell'ottimo globale: è possibile che l'arresto avvenga in punti di minimo/massimo locale, non trovando soluzioni migliori ad "un passo" di distanza (figura 3.1.3)⁽¹¹⁾. Tale metodologia è utile per diminuire i tempi computazionali, andando ad esaminare solo una porzione degli scenari possibili.

Un algoritmo greedy è particolare tipo di algoritmo euristico, dove si cerca una soluzione ottima attraverso la scelta della soluzione più "golosa" (proficua) ad ogni passo locale⁽¹²⁾.

Le euristiche sfruttate sono:

- Algoritmi greedy:
 - Partizionamento delle sedi sui veicoli;
 - Rimescolamento delle città in-day.
- Algoritmi di local search:
 - Reinsertion;
 - 2-opt;
 - Multi vehicle local search;

Ogni algoritmo prende in input un ordinamento nel formato di **route.csv**.

3.1.1 - Partizionamento

Questo algoritmo, per i giorni in cui sono disponibili due veicoli, effettua un partizionamento delle sedi; per ogni giornata in esame si otterranno due ordinamenti. Per ogni sede nella giornata si deve effettuare la scelta del veicolo da assegnarle:

- Nel caso in cui la sede in questione sia presente nell'ordinamento del primo veicolo la si assegna al secondo;
- Nel caso in cui la sede in questione sia presente nell'ordinamento del secondo veicolo la si assegna al primo;
- Se non è presente in nessuno dei due ordinamenti la si assegna a quello la cui ultima città, al momento della scelta, risulta più vicina.

Questo partizionamento mantiene l'ordinamento originale: non verranno effettuati scambi di posizioni, se due sedi sono consecutive nella schedulazione originale non potranno mai essere invertite nell'output, al più essere divise su due veicoli diversi.

3.1.2 - Rimescolamento greedy

Questo algoritmo effettua un partizionamento cercando però di migliorare la sequenza delle visite, modificando anche la posizione oltre che suddividendole. Per ogni giornata con un solo veicolo attivo:

- Si setta come ultima sede visitata e come primo elemento della schedulazione il deposito di partenza;
- Nella lista⁸ di destinazioni del giorno in questione si estrae la più vicina all'ultima sede visitata e la si inserisce nell'ordinamento del veicolo;
- Si aggiorna l'ultima sede visitata e si rimuove, dalla lista delle occorrenze ancora da schedulare, la sede appena inserita.
- Dopo aver inserito tutte le sedi si aggiunge in coda alla lista il deposito finale.

Per ogni giornata con due veicoli attivi:

- Per entrambi i veicoli si imposta come ultima sede visitata e come primo elemento dell'ordinamento il deposito di partenza (avremo due liste);
- Si estrae la sede più vicina per ognuna delle ultime sedi visitate e se ne calcola la distanza (temporale);
- La città corrispondente alla distanza minore viene inserita nel rispettivo ordinamento ed eliminata dalla lista delle sedi ancora da schedulare;
- Una volta assegnate tutte le sedi ad un veicolo si aggiunge in coda ad ognuno degli ordinamenti il rispettivo deposito finale.

3.1.3 - Reinsertion (RS)

Questo algoritmo effettua il miglioramento delle sequenze nei vari giorni per i due veicoli. Prende in input due schedulazioni (una per ogni veicolo) e cerca di migliorarle giorno per giorno tramite politica di reinserimento. Per ogni giornata di ogni schedulazione verrà effettuata una permutazione delle sedi, lasciando però in testa e in coda rispettivamente il deposito iniziale (comune) ed il deposito finale (del veicolo in questione).

⁸ Una collezione sequenziale di dati omogenei, facilmente estraibili grazie alla posizione e modificabili

Gli scambi di posizione delle sedi in ogni giornata verranno effettuate nel seguente modo:

- Inizia un ciclo “infinito” dove:
 - Si calcola il costo (temporale/spaziale) dell’attuale ordinamento e lo si salva per successivi confronti;
 - Per ogni città (esclusi deposito iniziale e finale) si cerca la posizione più vantaggiosa (esclusa prima e ultima) all’interno dell’ordinamento della giornata, effettuando uno spostamento in quel punto nel caso esista;
 - Una volta cercato di reinserire ogni città nella posizione migliore si calcola il costo di questo ordinamento: se migliore verrà mantenuto per il ciclo successivo, in cui si permuterà partendo da questa soluzione; in caso contrario si forzerà l’uscita dal ciclo “infinito”;

3.1.4 - 2-opt

Questo algoritmo prende in input due schedulazioni (una per veicolo) e lavora giornalmente. Si considera una schedulazione come una sequenza di nodi ed archi orientati, l’algoritmo effettua degli incroci nella rotta cambiando l’ordine di visita ai vari nodi. Con “incrocio” si definisce l’operazione di modifica della sequenza in cui: presi due archi orientati $A \rightarrow B$ e $E \rightarrow F$ presenti nel percorso si scambia la destinazione del primo con l’origine del secondo e viceversa ritrovandosi con i nuovi archi $A \rightarrow E$ e $B \rightarrow F$. Per mantenere il percorso praticabile bisognerà invertire la direzione di tutti gli archi del percorso compresi tra B ed E. Supponendo che nell’ordinamento iniziale la sequenza fosse $\dots A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \dots$ otterremo

$\dots A \rightarrow E \rightarrow D \rightarrow C \rightarrow B \rightarrow F \dots$ (figura 3.1.4.1)

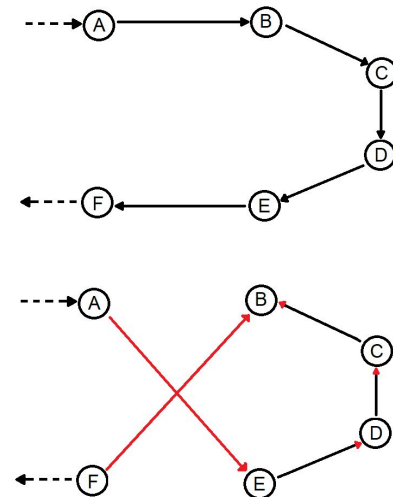


Figura 3.1.4.1 incrocio 2-opt

Più precisamente il 2-opt implementato avrà il seguente procedimento per ogni giornata:

- In un ciclo “infinito”:
 - Si calcola il costo dell'attuale schedulazione, lo si salva per successivi confronti e la si segna come “migliore” e come “migliore_locale”;
 - Per ogni giorno “i” nella schedulazione in esame, esclusi il primo e l'ultimo in quanto depositi ed il penultimo dato che porterebbe ad iterazioni non rilevanti:
 - Per ogni giorno “j” successivo ad “i” nella schedulazione in esame, esclusa l'ultima posizione dedicata al deposito:
 - Si effettua l'incrocio degli archi $(i, i+1)$ e $(j, j+1)$;
 - Si calcola il costo della nuova schedulazione e se più vantaggioso del costo della soluzione “migliore_locale” si segna quella appena calcolata come tale;
 - Infine si confronta la soluzione “migliore_locale” con la “migliore” nel caso in cui la prima presenti un costo più vantaggioso si modifica la schedulazione da analizzare con quella “migliore_locale” che diventa anche “migliore”, altrimenti si forza l'uscita dal ciclo “infinito”.

Per ogni esecuzione del ciclo più esterno si otterrà il migliore incrocio sulla schedulazione in esame, se trovato viene effettuata una successiva iterazione, altrimenti viene restituita la soluzione migliore trovata tra quelle esaminate.

3.1.5 - Multi Vehicle Local Search (MVLS)

I precedenti algoritmi lavorano sulle singole schedulazioni giornaliere, a differenza del Multi Vehicle Local Search (MVLS) che gestisce anche uno spostamento delle sedi tra le varie giornate e tra i due veicoli. Prende in input le schedulazioni dei due veicoli per rimescolare le sedi cercando possibili miglioramenti inter-giorno.

Si basa sul concetto descritto nell'algoritmo di reinsertion precedentemente descritto:

- Per ogni giornata (g) di ogni veicolo (v) viene estratta la sequenza delle sedi da visitare, esclusi deposito di partenza e arrivo;
 - Per ogni sede nella sequenza estratta si cerca la miglior posizione per la biblioteca in esame negli ordinamenti che non presentano già un'occorrenza della sede in questione (figura 3.1.5.1), con giorno diverso

da “g” e veicolo diverso da “v” (per non inserirlo nella schedulazione di partenza o in un ordinamento con la sede già presente);

- Si prende in considerazione la schedulazione che, dopo l’inserimento della sede, presenta il costo minore e, se l’aumento di quest’ultimo non supera il miglioramento della schedulazione in esame (senza la sede in questione), si salva il risultato per i successivi cicli, altrimenti si mantengono gli ordinamenti precedenti alla fase di estrazione e inserimento;
- Vengono restituite le due schedulazioni;

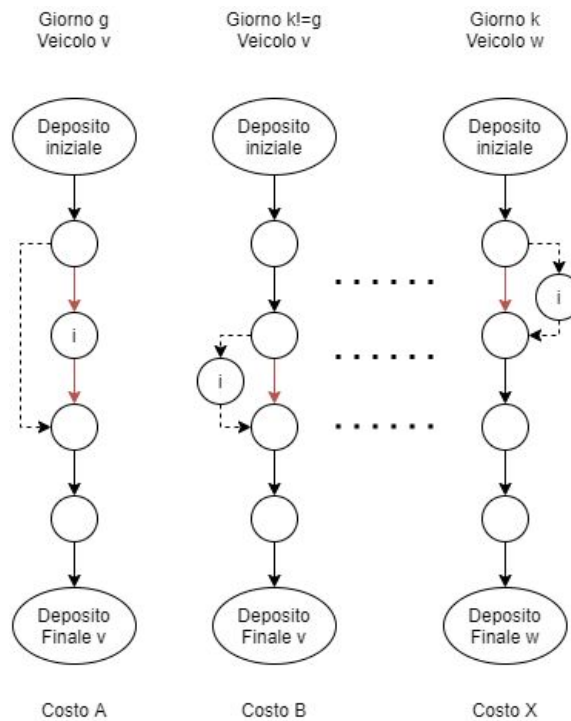


Figura 3.1.5.1 fase di estrazione e reinserimento, con controllo dei costi

3.2 - Descrizione tecnica

Tutti gli algoritmi implementati, i test, le misurazioni e visualizzazioni sono stati computati su un desktop con le seguenti specifiche:

- Processore AMD A10-7850K Radeon R7, 12 Core, 3.7 GHz;
- Ram 16 GB;
- Sistema operativo windows 10, 64 bit.

In seguito alla fase di definizione del modello e quella di costruzione degli algoritmi euristici, segue la fase di concretizzazione degli stessi con l'uso di un linguaggio di programmazione.

Questa parte si è suddivisa in tre fasi:

- 1) analisi preliminare;
- 2) definizione delle strutture dati;
- 3) stesura del codice.

Lo scopo dell'analisi preliminare è quella di individuare le migliori scelte progettuali e il linguaggio di programmazione che meglio si adatta all'implementazione degli algoritmi. La prima operazione è stata la comprensione dei dati forniti dalla rete bibliotecaria e il loro formato:

- **Orario bibliobus.docx**: rappresenta una tabella contenente i giorni lavorativi e per ognuno la sequenza di sedi da visitare, rappresentate dal loro nome o città di appartenenza; per ogni sede l'orario indicativo in cui deve essere servita;
- **Indirizzi_biblio_21-10-2019.ods**: è un elenco delle biblioteche presenti sul territorio specificando con relative informazioni (nome, comune, indirizzo ecc.);
- **request_cr.csv**: contiene un elenco di prestiti dal 2016 al 2018 con il numero di ordine, la data e la coppia di biblioteche interessate.

Dopo un'attenta valutazione del contenuto di questi file, si è resa necessaria una "pulizia" dei dati e l'estrapolazione delle informazioni strettamente necessarie per la costruzione del modello: per esempio, alcuni prestiti registrati in **request_cr.csv** sono tra una biblioteca e sé stessa, oppure in **Indirizzi_biblio_21-10-2019.ods** sono specificate biblioteche non calendarizzabili.

Una volta effettuato il filtraggio si è svolta la fase di manipolazione dei dati “puliti” in modo che risultassero pratici. In questa parte la trasformazione ha prodotto i seguenti file:

- **CitiesLocation.csv**: contiene le coordinate geografiche di ogni biblioteca con relativo nome e indirizzo;
- **Route.csv**: contiene, per ogni giorno, il solo ordine delle biblioteche da visitare;
- **Route_time.csv**: contiene, per ogni giorno, il solo orario in cui ogni biblioteca in **Route.csv** deve essere servita..

Come si può osservare il primo file è stato ricavato da **Indirizzi_biblio_21-10-2019.ods**, mentre gli ultimi due da **Orario bibliobus.docx**. Un'altra scelta importante è stata quella di uniformare il formato dei file in **csv** data l'alta portabilità e la facilità di lettura.

Ottenuta una tipologia di dati più uniforme e di utilità pratica, è stato possibile passare alla fase di creazione delle componenti del modello e delle strutture in cui contenerle.

Nel file **CitiesLocation.csv** sono contenute le geolocalizzazioni delle sedi, grazie a queste e a servizi di routing (**WazeRouteCalculator**) è possibile estrarre i costi sede-sede dai seguenti file:

Nel file **CitiesLocation.csv** sono contenuti i nodi, quindi si è reso necessario la “sola” definizione degli archi del grafo e quindi i seguenti file:

- **Corrected_times_computed.csv**: una matrice che descrive i costi (in termini di tempo) per ogni coppia di città;
- **Distances_computed.csv**: una matrice che descrive i costi (in termini di chilometri) per ogni coppia di città.

Entrambi i file presentano una matrice 64×64 in cui per ogni cella della riga **r** della biblioteca **i** è presente il tempo/chilometraggio per raggiungere la biblioteca corrispondente alla colonna⁹.

⁹ La natura della matrice implica che per ogni $i = j$, il costo sia nullo.

Prima di effettuare la stesura del codice, è stata affrontata la scelta del linguaggio di programmazione considerando: la facilità nella manipolazione dei dati, la performance necessarie in fase di esecuzione, la semplicità nel trattamento/analisi dei risultati e della stesura del codice.

Essendo la parte di gestione dati più rilevante rispetto alla performance esecutiva si è optato per un linguaggio ad alto livello che permettesse una stesura del codice semplificata: **Python**.

Per la modellizzazione del problema di distribuzione si è scelto **AMPL**, un linguaggio appositamente creato per descrivere problemi di programmazione lineare. Essendo interpretabile da vari risolutori è risultato la scelta migliore.

Per gestire al meglio il codice, le varie versioni di quest'ultimo, le librerie e per la risoluzione dei modelli di programmazioni sono stati sfruttati i seguenti tool:

- **Anaconda**⁽¹³⁾: è una piattaforma di distribuzione dei linguaggi di programmazione Python e R per il calcolo scientifico (data science, applicazioni di machine learning, elaborazione dati su larga scala, analisi predittiva, ecc.), che mira a semplificare la gestione e la distribuzione dei pacchetti;
- **Jupyter Notebook**⁽¹⁴⁾: è un'applicazione web open source che consente di creare e condividere documenti che contengono codice live, equazioni, visualizzazioni e testo narrativo. Gli usi includono: pulizia e trasformazione dei dati, simulazione numerica, modellazione statistica, visualizzazione dei dati, apprendimento automatico e molto altro;
- **GitHub**⁽¹⁵⁾: è un servizio di hosting per progetti software. Il nome deriva dal fatto che **GitHub** è una implementazione dello strumento **Git**;
- **Gurobi Optimize**⁽¹⁶⁾: Si tratta di un risolutore ottimizzato per la risoluzione di problemi di programmazione lineare.

Di seguito sono riportate le principali librerie utilizzate nella stesura del codice:

- **Geopy.geocoders**: questa libreria dà la possibilità di ricavare le posizioni geografiche di località specificate con il rispettivo nome o indirizzo e viceversa. Si è resa particolarmente utile in fase di costruzione del file **CitiesLocation.csv**;
- **WazeRouteCalculator**: permette di calcolare il percorso migliore tra due coordinate geografiche con il relativo costo temporale e spaziale, utilizzato per creare le matrici di costo contenute nei file **Corrected_times_computed.csv** e

Distances_computed.csv. Restituisce anche una lista di coordinate che permettono di ricostruire il percorso e visualizzarlo su mappa;

- **Folium:** permette l'utilizzo di **leaflet.js**, una libreria javascript per le mappe interattive, con semplici funzioni altamente personalizzabili. Tramite **geopy** e **WazeRouteCalcolator** è possibile visualizzare le posizioni delle biblioteche e i percorsi utilizzati;
- **Ipynb:** durante la stesura del codice è stata fondamentale la scrittura di funzioni riutilizzabili, questa libreria permette di creare un file di funzioni e importarle in altri file ipynb in **Jupyter**;
- **Pandas:** libreria per la gestione ed analisi di dati tramite strutture e funzioni apposite. Le strutture più usate sono state DataFrame e Series per poter rappresentare in maniera comoda e robusta i dati a nostra disposizione, mentre le funzionalità più utili sono state la gestione delle due strutture e la facilità di lettura e scrittura nei vari file;
- **Pandas_ods_reader:** espansione di pandas, fornisce funzioni per la lettura e scrittura di file nel formato “ods” restituendo un oggetto appartenente a pandas per sfruttare il suo panorama di funzionalità;
- **Numpy:** fornisce strutture dati (matrici ed array multidimensionali) e funzioni ottimizzate per la loro gestione, migliorando le prestazioni dei calcoli tra dati in queste strutture;
- **Math:** fornisce funzioni per la gestione dei numeri e particolari operazioni su di essi;
- **Datetime:** l'utilizzo di questa libreria fornisce supporto per operazioni e formattazione di orari, date e, in generale, tempistiche. Sfruttata per gestire gli orari e le date degli ordini, anche trasformando le date in giorni della settimana;
- **Func tools:** permette di creare oggetti richiamanti funzioni, anche con argomenti predefiniti (anche solo parzialmente), per renderne più semplice l'utilizzo;
- **Csv:** manipolazione dei file csv;
- **Tqdm:** funzionalità estetica, per il controllo del livello di progressioni delle iterazioni;

- **Mathplotlib.pyplot**: libreria contenuta in **matplotlib**, sfrutta le strutture presenti in **numpy** e permette di visualizzarne una vasta tipologia di grafici altamente personalizzabili ed integrabili con librerie esterne;
- **Ast**: Abstract Syntax Trees, fornisce funzionalità per processare alberi e interpretare i valori presenti. In particolare viene usato per “parsare” un formato di file (**Frequencies.csv**) contenente liste di liste, non direttamente interpretabili dal lettore di csv;
- **random**: libreria sfruttata per generare le schedulazioni casuali.

Il progetto è diviso in due macro parti:

- Algoritmi;
- Dati.

Nella prima sono presenti le implementazioni delle euristiche precedentemente descritte e alcuni programmi per la visualizzazione e riassunzione dei dati:

- **SimpleRouting.ipynb**: in questo file è presente il codice per la ricostruzione delle route e dell'algoritmo greedy;
- **LocalSearch.ipynb**: qui si trova l'algoritmo di *reinsertion*;
- **2-opt.ipynb**: contiene l'algoritmo 2-opt;
- **multi-vehiclelocalsearch.ipynb**: in questo file è presente il codice del multi vehicle local search;
- **functions.ipynb**: un file riassuntivo delle funzioni più utilizzate e comuni;
- **Geolocation.ipynb**: contiene le funzionalità per la localizzazione delle città e calcolo delle matrici delle distanze;
- **MapLocator.ipynb**: permette la visualizzazione su mappa delle città e dei percorsi delle varie schedulazioni;
- **orders_analysis.ipynb**: codice per elaborare lo storico ordini, suddividendolo per giornata e filtrando ordini non considerabili;
- **Resume_table.ipynb**: aggregazione dei parametri temporale e spaziale in tabelle per migliorarne la gestibilità;
- **Random.ipynb**: file per la generazione della schedulazione randomica;
- **Test.ipynb**: file di testing per alcune parti di codice;
- **schedule_comparison.ipynb**: file per la comparazione di tutte le schedulazioni.

Nella seconda parte sono presenti cartelle contenenti i file dei dati relativi a diversi aspetti del progetti divisi per categoria:

- Cartella **distanze**: contiene i file relativi alle distanze temporali/spaziali e le frequenze degli ordini tra le varie città;
- Cartella **maps**: contiene le rappresentazioni dei vari percorsi e localizzazioni delle sedi trattate;
- Cartella **orari**: contiene i file della schedulazione originaria fornita da RBC;
- Cartella **turni_f**: contiene le varie schedulazioni ottenute dalle euristiche e dai modelli;
- Cartella **altro**: contiene file di varia natura non aggregabili nelle altre cartelle, come la soluzione dei modelli OSS e FOSS, i modelli in un formato (.MOD) interpretabile a Gurobi (compresi i loro parametri) e le tabelle riassuntive di tempo/spazio delle varie schedulazioni;
- Cartella **explorative_analysis**:
 - Cartella **FrequenciesResults.ipynb**: vari file intermedi utilizzati per l'analisi delle frequenze, grafici comparativi e file riassuntivo sul grado di servizio nei vari ordinamenti generati;
 - Cartella **distribution**: contiene i file derivanti dall'analisi di distribuzione degli ordini;
 - File **Comparison_chart.ipynb**: file relativo al riassunto dei dati e loro visualizzazione;
 - File **Distribution_analysis.ipynb**: file contenente l'analisi della distribuzione degli ordini;
 - File **Sol_file_interpretation.ipynb**: permette l'interpretazione dei risultati restituiti da Gurobi per la risoluzione del modello;
 - File **Waiting_days_dat.ipynb**: file per generare i parametri di input a gurobi per la risoluzione dei modelli;
 - File **explorative_analysis.ipynb**: file contenente l'analisi esplorativa;

Di seguito sono riportati gli alberi di generazione dei vari file:

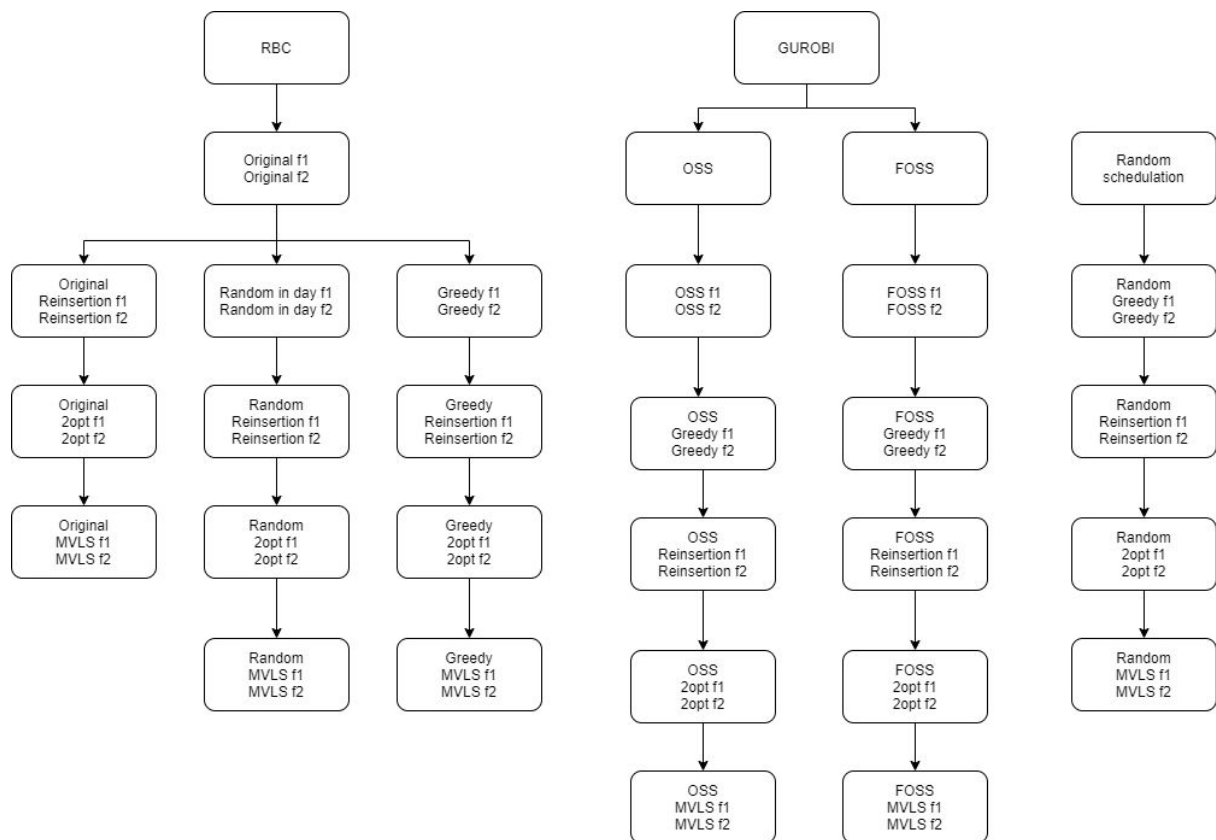


Figura 3.2.1 Alberi rappresentanti la generazione delle schedulazioni

Va ricordato che il modello dato a Gurobi per la generazione delle schedulazioni **OSS** e **FOSS** è dipendente dallo storico degli ordini; quindi è necessaria una fase di estrazione dei parametri interessati prima di poter generare le soluzioni.

Capitolo IV: Analisi sperimentale

In questo capitolo verranno presentati i risultati degli esperimenti effettuati sulle varie schedulazioni e sui dati in nostro possesso. In particolare sarà analizzata la distribuzione degli ordini (ottenuti dallo storico) secondo vari livelli di aggregazione e verranno confrontati i risultati delle schedulazioni secondo i KPI definiti nel capitolo 2.

4.1 - Distribuzioni degli ordini

Avendo basato i due modelli per l'ottimizzazione della distribuzione delle sedi (OSS e FOSS) sullo storico degli ordini risulta opportuno analizzare come la distribuzione degli ordini si presenta sotto diverse forme di aggregazione.

Tutte le aggregazioni sono effettuate partendo dal file “**request_cr.csv**” descritto nel paragrafo 1.1.3.

4.1.1 - Aggregazione degli ordini per biblioteca e per data

In questa sezione, l'aggregazione viene fatta a due livelli: di biblioteca e di data. Questo permetterà di capire come varia il volume dei prestiti e quali sono i volumi più correnti nello storico. Con il termine volume viene indicata un ordine effettuato, secondo il livello di aggregazione, contenente tot materiale e.g. in questa sezione si avrà un volume per ogni biblioteca e data.

Nella figura 4.1.1.1 viene presentata la distribuzione degli ordini in entrata alle biblioteche in due grafici, quello a sinistra mostra sull'asse delle ascisse il volume di un ordine e sull'asse delle ordinate il numero di volte in cui compare quel dato volume. Il grafico a destra visualizza la funzione di probabilità cumulativa, ovvero per una data x la probabilità (y) che un ordine di volume minore o uguale alla x in questione venga ricevuto.

Si nota una densità maggiore sui volumi tra 0 e 15, in cui ricade circa il 90% della probabilità del volume di un ordine. All'aumento del volume si incontrano valori meno presenti, come si vede dal grafico di destra la probabilità di avere volumi degli ordini maggiori di 20 è davvero bassa. Questo indicherà che un dato giorno è difficile ricevere ordini superiori alle 20 unità di materiale, suggerendo che in una giornata solitamente si registrano al massimo 15 prestiti.

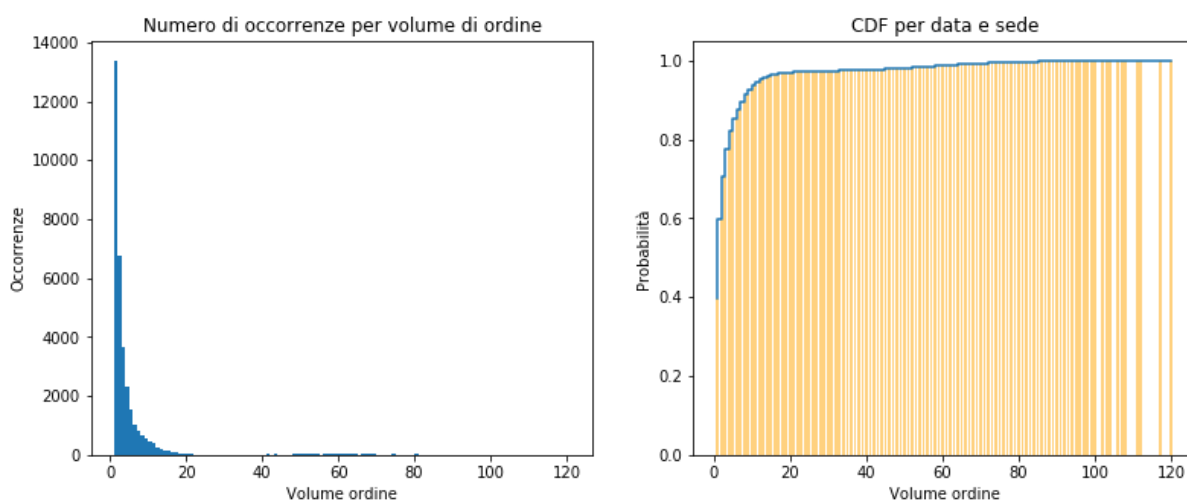


Figura 4.1.1.1 Distribuzione generale degli ordini in entrata per sede e data

Risulta opportuno analizzare se l'andamento generale riportato nella figura soprastante rispecchia quello di tutte le biblioteche presenti nello storico. Per questo sono state estratti due rappresentanti per le principali categorie di sedi presenti: una sede importante che dovrebbe presentare un numero maggiore di ordini e anche di volumi, ed una sede minore, che a causa della minor densità di popolazione nel suo territorio e minor dimensioni dovrebbe presentare volumi inferiori. La figura 4.1.1.2 visualizza le distribuzioni di questi due campioni: i due grafici nella parte superiore raffigurano la distribuzione dei prestiti in una sede di maggior importanza, i due nella parte inferiore la distribuzione di una sede minore. Come previsto la curva della funzione di distribuzione per la sede maggiore è meno ripida, poiché ha più volumi di prestiti con occorrenze simili.

La seconda distribuzione invece rispecchia meglio l'andamento generale, con un picco nei primi valori ed una salita molto blanda all'aumentare del volume dei prestiti.

La distribuzione generale è più simile all'andamento delle sedi minori data la loro maggior presenza; esistono meno biblioteche di grande importanza a fronte di un alto numero di biblioteche minori. La presenza della prima categoria può spiegare perché si ha una curva più arrotondata nella funzione di probabilità generale, smorzando la ripidità presentata dalla seconda categoria.

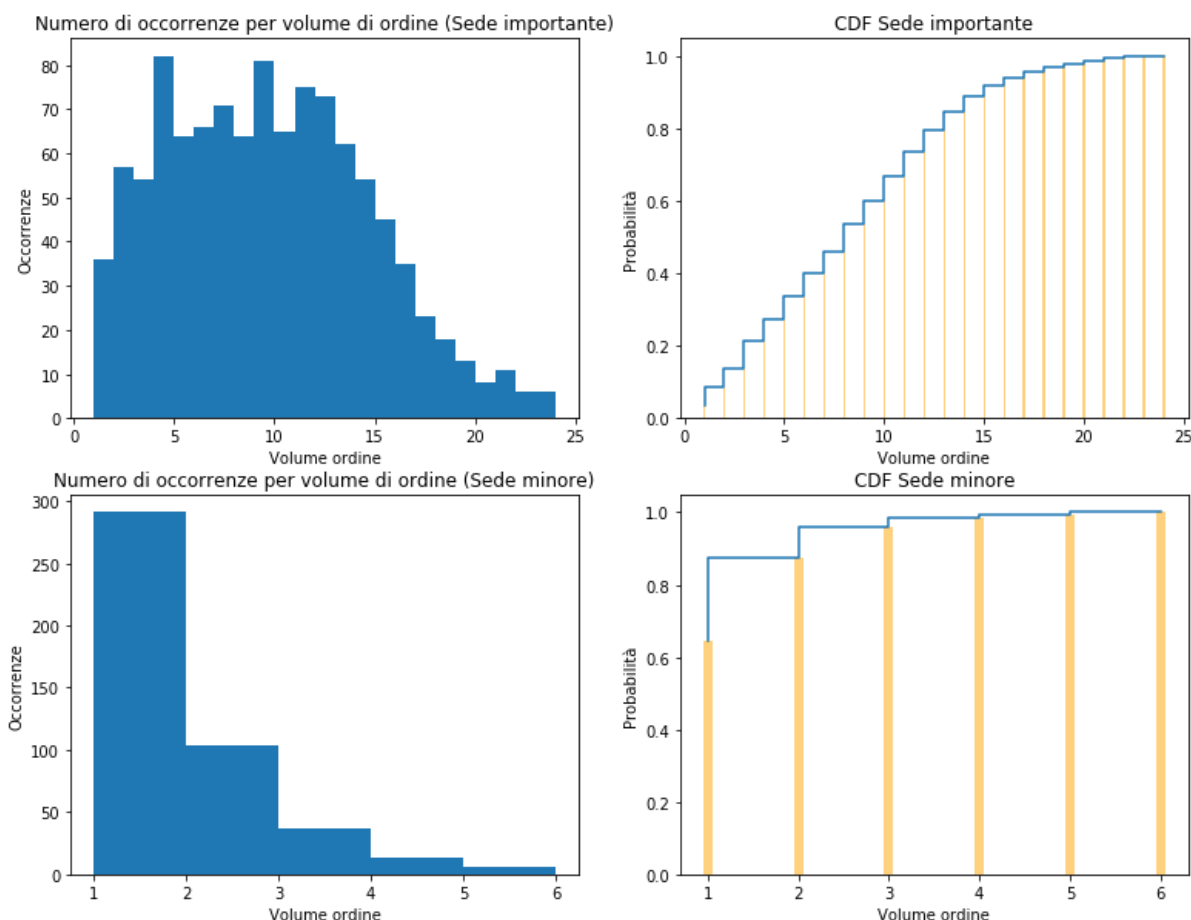


Figura 4.1.1.2 Confronto delle distribuzioni di due sedi. Grande sopra, minore sotto

Esiste una sede che non può essere pienamente rappresentata dalle categorie di “sede maggiore” e “sede minore”, ovvero quella del punto di scambio con la rete interbibliotecaria di Brescia. La distribuzione raffigura una varietà nel volume degli ordini molto più ampia che nei casi precedenti, inoltre i picchi avvengono non più tra i primi valori, ma più spostati verso il centro (figura 4.1.1.3). La sua presenza non influenza eccessivamente l'andamento generale dato che, come nel caso delle biblioteche maggiori, non riesce a compensare il grande numero di biblioteche minori. Questa distribuzione influirà in minima parte nell'arrotondamento della curva di crescita nella funzione di probabilità (figura 4.1.1.1).

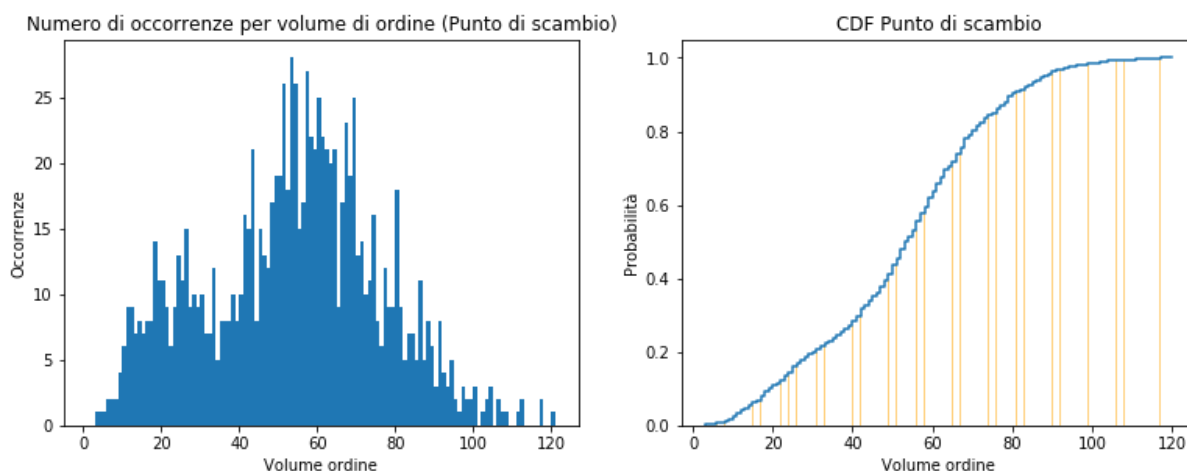


Figura 4.1.1.3 Unica distribuzione che si discosta, il punto di scambio

Viene ora presentata la distribuzione generale degli ordini in uscita, ovvero i prestiti richiesti. A differenza dell'andamento generale degli ordini in entrata (richieste di prestito ricevute, figura 4.1.1.1), quasi la totalità degli ordini è di volume molto ridotto (1 o 2 unità di materiale); per questo la curva della funzione di probabilità cumulativa risulta molto ripida nei primi valori, sottolineando che in un dato giorno è quasi certo che avvengano prestiti di numerosità 1 (figura 4.1.1.4).

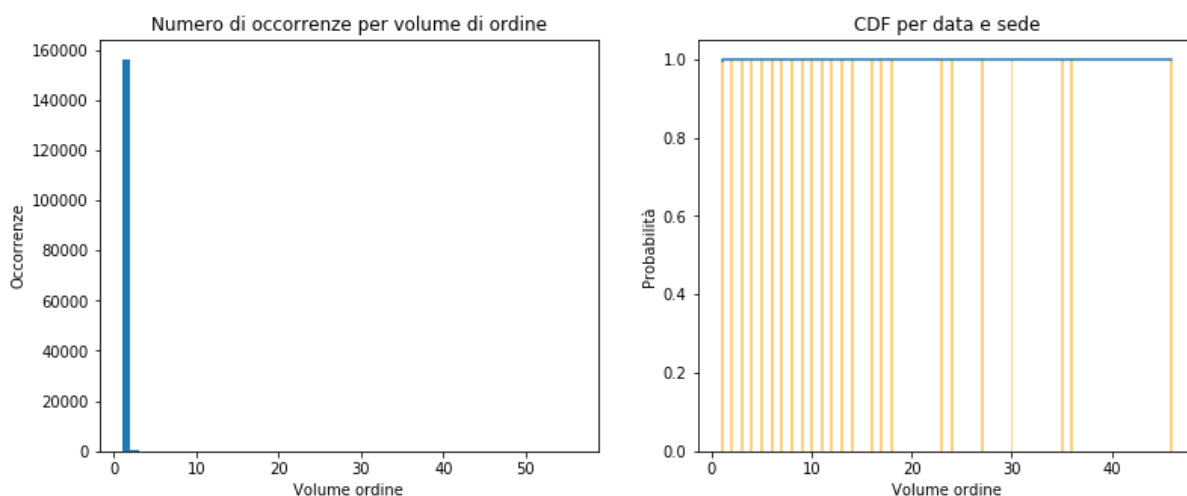


Figura 4.1.1.4 Distribuzione generale degli ordini in uscita per sede e data

Anche analizzando la distribuzione delle richieste di prestito in biblioteche maggiori e minori (figura 4.1.1.5) viene rispecchiato l'andamento generale, con un elevato numero di volume ridotto un dato giorno.

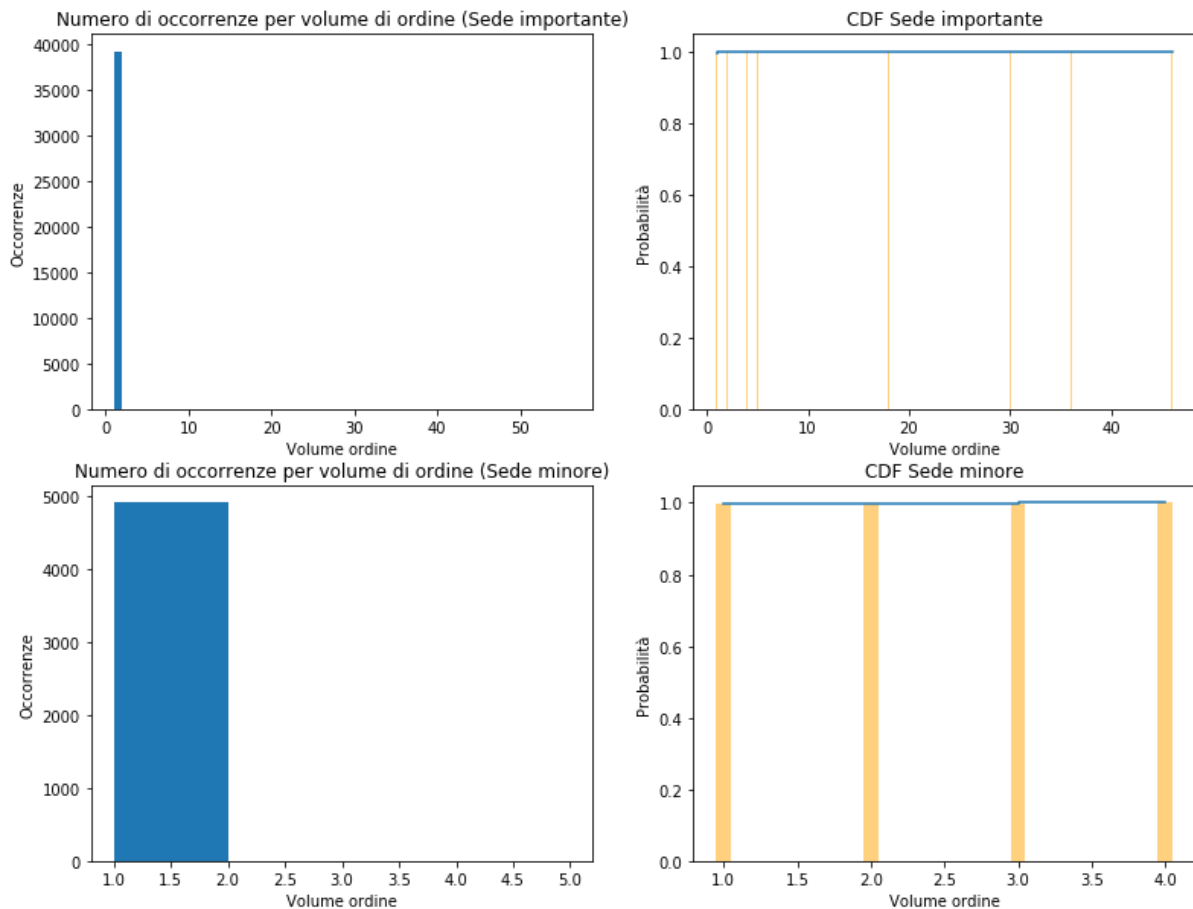


Figura 4.1.1.5 Unica distribuzione che si discosta, il punto di scambio

Anche per il punto di scambio con Brescia (figura 4.1.1.6) il risultato raffigura precisamente l'andamento generale, non pervengono outlier analizzando gli ordini in uscita.

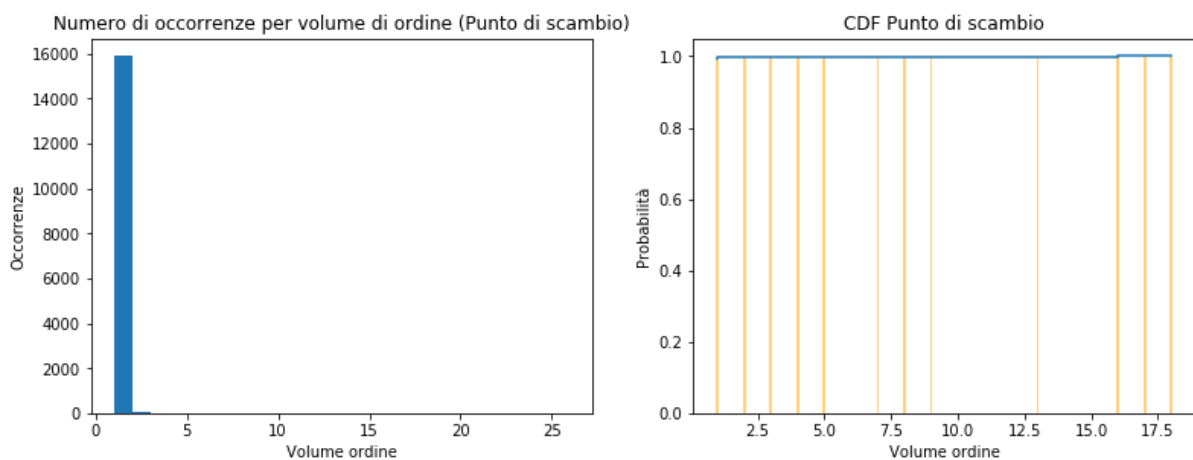


Figura 4.1.1.6 Unica distribuzione che si discosta, il punto di scambio

In conclusione, i volumi delle richieste in entrata risultano più variegati (anche a volumi maggiori) soprattutto per le sedi più grandi, mentre alle sedi minori verranno

richiesti volumi molto più contenuti. Al contrario tutte le sedi effettuano molte più richieste di prestiti con volumi contenuti o addirittura unitari.

4.1.2 - Aggregazione giorno e coppia di sedi

In questa sezione gli ordini saranno raggruppati per giornata in cui vengono effettuati (lunedì, martedì, mercoledì, giovedì, venerdì, sabato e domenica) e per coppia di biblioteche interessate dallo scambio.

In tutte i grafici sono stati esclusi degli outlier: nei grafici presenti nella figura 4.1.2.1 vengono estromessi 5 volumi molto alti e con occorrenze singole, questi avrebbero reso i due grafici molto meno leggibili, schiacciandoli sulla sinistra; per gli stessi motivi, in ogni giornata presentata nella figura 4.1.2.2 sono stati rimossi volumi eccessivamente alti ad una sola occorrenza.

La figura 4.1.2.1 mostra, come nel precedente caso di aggregazione, una densità maggiore di occorrenze per i volumi minori. Nel grafico a destra, che rappresenta per ogni valore sull'asse delle ordinate la probabilità che un ordine di volume x sia effettuato tra due biblioteche, viene evidenziata una salita meno arrotondata rispetto a quella mostrata nella figura 4.1.1.1, seppur simile. Quasi la totalità degli ordini viene fatta con volumi ridotti, riducendo la rapidità di salita fino ad avere solo qualche sporadico ordine mono occorrenza di notevoli dimensioni.

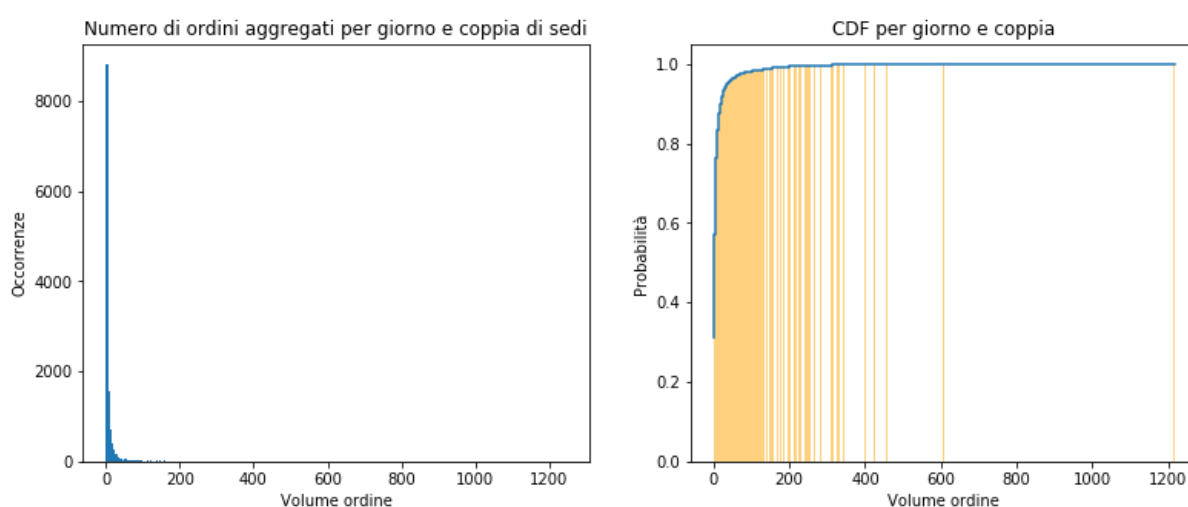
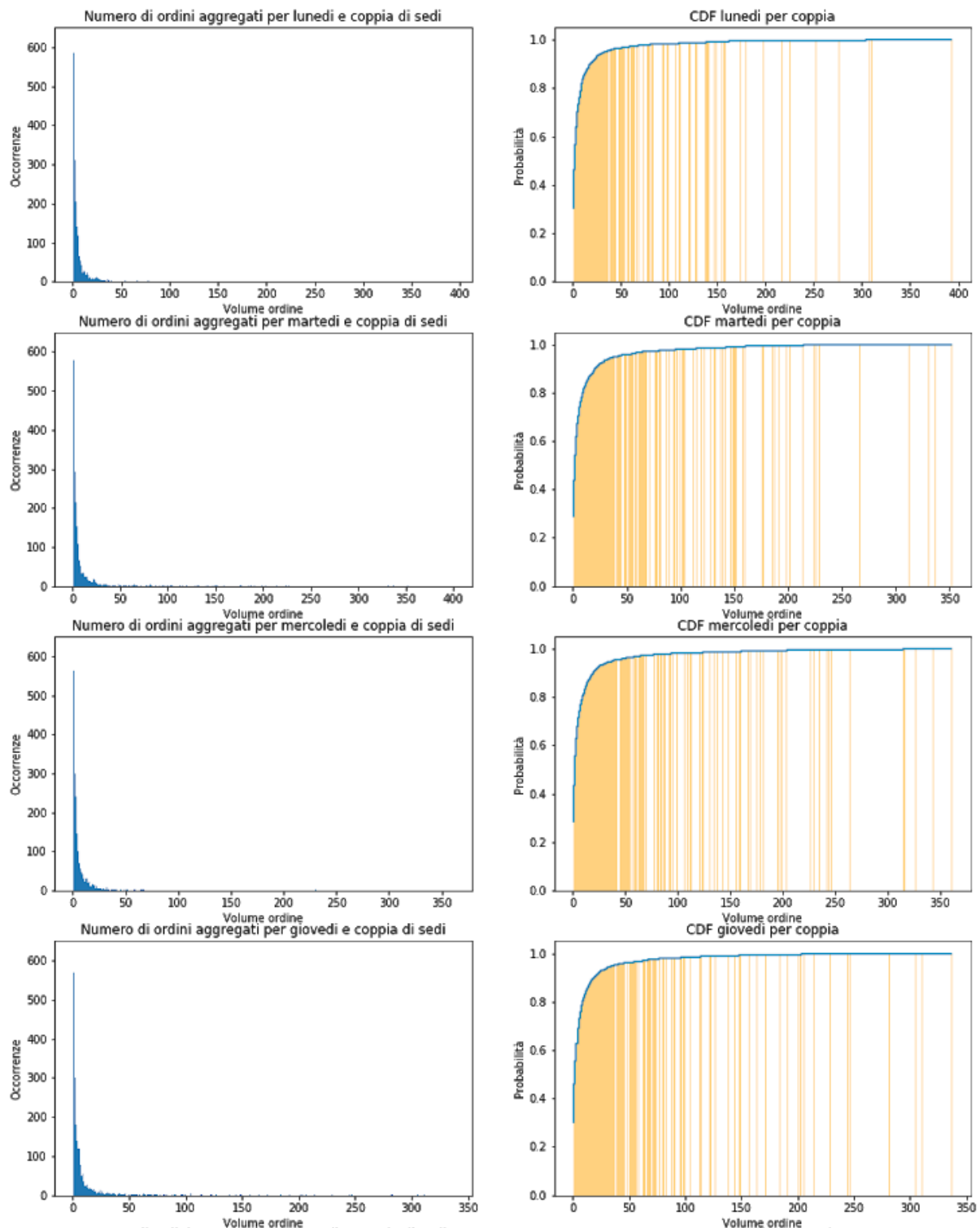


Figura 4.1.2.1 Distribuzione del volume degli ordini raggruppati per giorno e coppia di sedi

Nella figura 4.1.2.2 vengono presentate le distribuzioni per ogni giornata. Queste sono quasi del tutto simili all'andamento della funzioni di probabilità del grafico generale, non vi sono casi particolari che vanno a modificare la forma. Dagli istogrammi nella

parte di sinistra delle coppie si può vedere che gli ordini sono divisi diversamente tra le giornate, dato evidenziato nella figura 4.1.2.3, questo farà pesare di più la forma delle distribuzioni nei giorni più popolosi rispetto a quella dei giorni meno popolati. Avendo tuttavia andamenti molto simili nelle funzioni di probabilità, il risultato (figura 4.1.2.1, parte destra) rispecchia fedelmente tutte le distribuzioni giornaliere presentate.



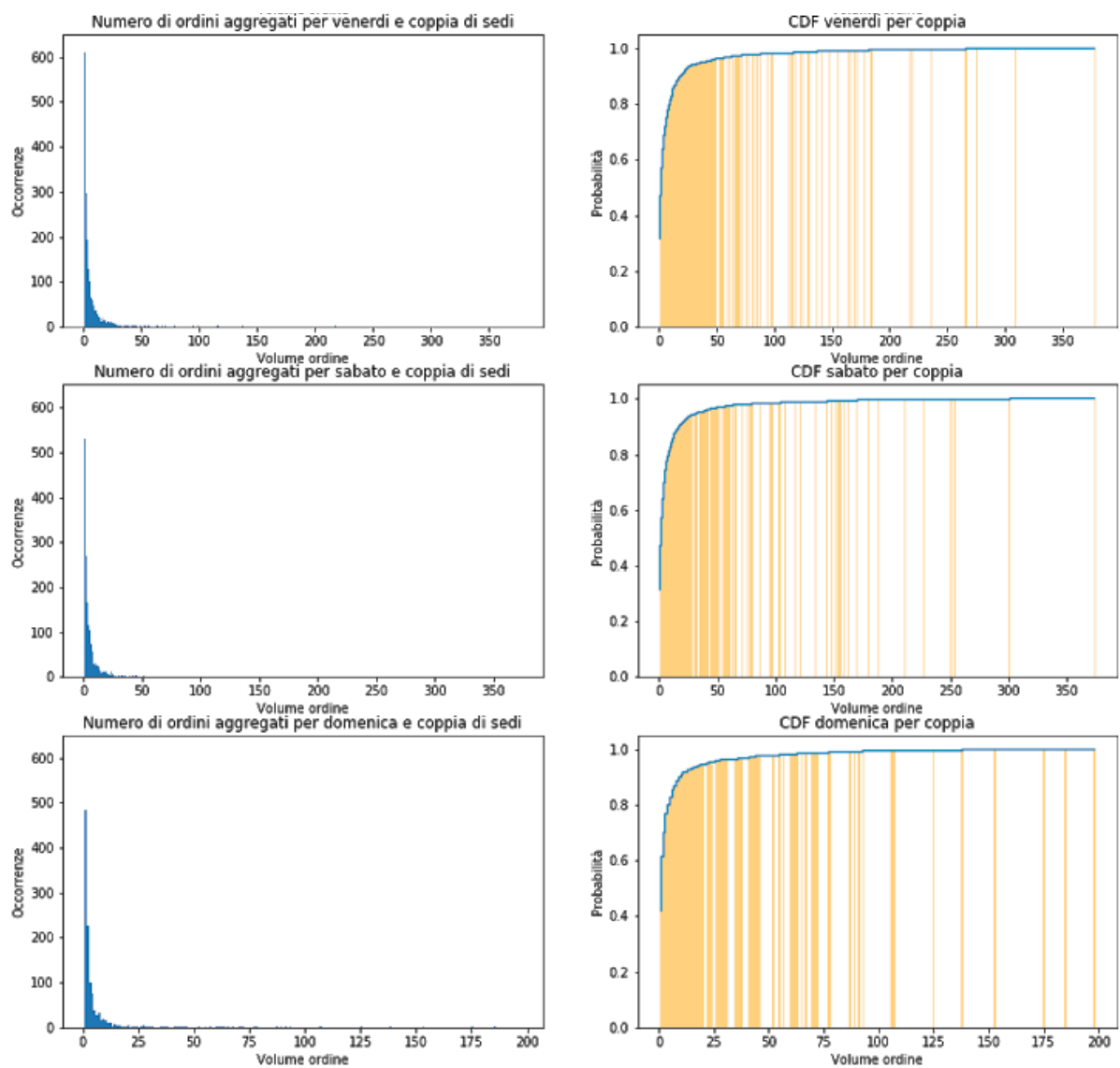


Figura 4.1.2.2 Distribuzione del volume degli ordini per ogni giorno della settimana

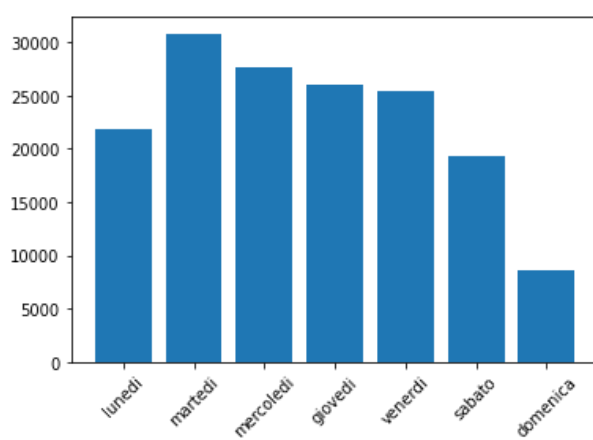


Figura 4.1.2.3 Numerosità degli ordini nelle varie giornate

4.1.3 - Aggregazione per coppia di biblioteche

In questa sezione lo storico degli ordini sarà aggregato solamente per coppia di biblioteche ed il numero di ordini effettuati da una all'altra. Per una visualizzazione migliore sono stati eliminati alcuni valori eccessivamente alti (10 volumi mono occorrenza).

Viene raffigurato l'andamento evidenziato nelle precedenti sezioni, ovvero una densità maggiore di ordini nei primi valori, anche se in questo caso meno accentuata.

La probabilità di avere un volume di ordini minori di 200 risulta molto alta, quindi si può affermare che le richieste di scambio difficilmente avranno volumi oltre questa soglia.

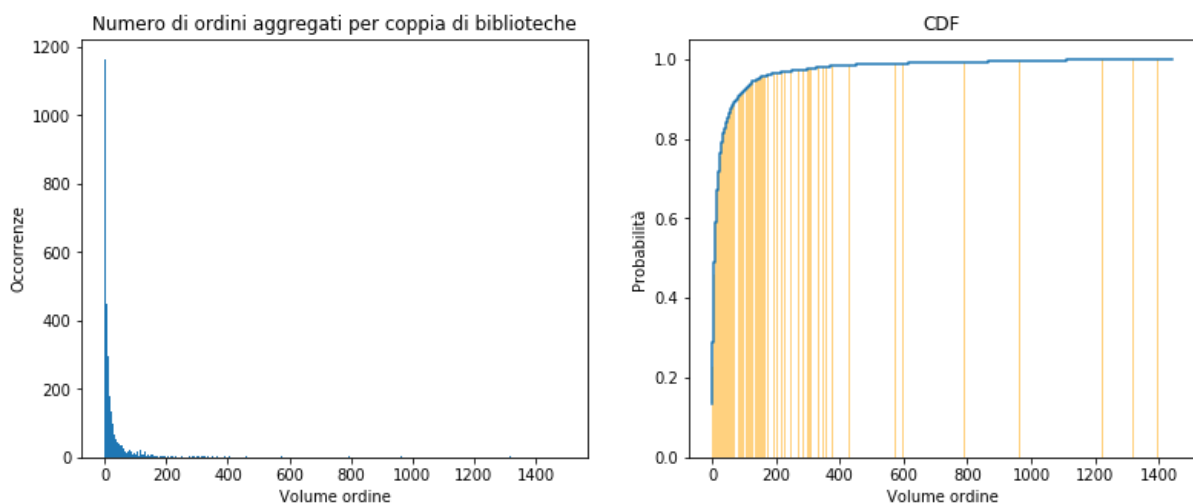


Figura 4.1.3.1 Distribuzione del volume degli ordini per coppie di biblioteche

4.1.4 - Considerazioni finali

Dai precedenti esperimenti è emersa una tipologia di distribuzione simile per ogni livello di aggregazione (se non per un singolo caso particolare 4.1.1.2), questo permette di affermare che i parametri estratti dallo storico (**request_cr.csv**), utilizzati nei modelli OSS e FOSS, rispecchiano pienamente la distribuzione degli ordini, nonostante la loro aggregazione.

4.2 - Valutazione delle soluzioni

In questa sezione verranno analizzati i risultati delle schedulazioni ottenute. Queste ultime saranno valutate grazie ai KPI precedentemente descritti:

- [1] Tempo di percorrenza giornaliera, per veicolo e totale definito **tempo**;
- [2] Distanza totale giornaliera, per veicolo e totale definita **distanza**;
- [3] Bilanciamento medio del lavoro tra i due veicoli in uso definito **carico**;
- [4] Tempi medio di attesa prima della presa in carico di un ordine definito **mp**;
- [5] Tempo medio di attesa dalla richiesta di prestito alla relativa consegna definito **ms**.

I KPI **tempo** [1], **distanza** [2] e **carico** [3] sono applicati ad ogni schedulazione dato che ogni cambiamento negli ordinamenti giornalieri può far variare questi indicatori.

I KPI **mp** [4] e **ms** [5] saranno applicati solamente nelle schedulazioni in cui sussiste una possibile differenza della distribuzione delle sedi nelle giornate (schedulazione di partenza e la risultante dal multi vehicle local search); il riordinamento interno ai vari giorni non cambia il tempo di attesa per il ritiro o il completamento di un ordine (greedy, reinsertion e 2opt non effettuano spostamenti inter giornalieri).

Sono state introdotte due basi casuali:

- Random In Day: le occorrenze delle sedi nei vari giorni sono uguali a quelle della schedulazione originale, tuttavia l'ordinamento giornaliero viene ricombinato casualmente prima di applicare le euristiche;
- Random Schedule: si distribuiscono casualmente le sedi nell'arco della settimana, mantenendo però il numero di occorrenze per ogni sede ed alcuni vincoli di ammissibilità (lo stesso giorno, anche in due veicoli diversi, non può essere presente più volte la stessa sede). Questo genererà una schedulazione casuale, non creata per essere già performante.

La prima verificherà l'efficacia degli algoritmi di ottimizzazione giornalieri (greedy, reinsertion e 2opt), verificando che la loro applicazione porti a ordinamenti con tempistiche simili a quelle basate su schedulazioni già pensate per essere buone in partenza (base originale e greedy).

La seconda verrà confrontata con gli ordinamenti generati dai modelli OSS e FOSS, verificando se la costruzione di un'assegnazione ottimale è utile per ottenere soluzioni migliori.

4.2.1 - Soluzioni derivanti dalla schedulazione fornita

In questa sezione saranno confrontati i risultati delle euristiche derivanti dalla schedulazione originaria presentata dalla rete bibliotecaria cremonese; queste soluzioni sono state ottenute mantenendo l'assegnamento originario delle sedi ai vari giorni come base da cui partire. Applicando prima diversi metodi di ordinamento nella giornata al fine di migliorare le tempistiche si arriva all'applicazione del multi vehicle local search, in cui le sedi verranno anche riposizionate tra i giorni e tra i veicoli a disposizione. In riferimento all'figura 3.2.1 vengono presentate le soluzioni derivanti dalla radice "RBC".

Di seguito sono riportate le tabelle riassuntive delle schedulazioni basate su schedulazione originale (figura 4.2.1.1), randomizzata giornalmente (figura 4.2.1.2) e con applicato algoritmo greedy (figura 4.2.1.3). Su ognuna di queste sono stati applicati in sequenza: reinsertion, 2opt e MVLS; per ogni schedulazione risultante sono stati ricavati i KPI di **tempo** [1], **distanza** [2] e **carico** [3]. Si vorrà verificare l'efficacia delle euristiche nell'abbattimento dei costi e controllare se questi miglioramenti penalizzano altri fattori.

Original	Base	RS	2opt	MVLS
tempo	1777.0	1568.0	1434.0	1399.0
distanza	1691.0	1465.0	1309.0	1218.0
carico	0.25	0.18	0.1	0.45

Figura 4.2.1.1 Tabella riassuntiva per derivanti della schedulazione originale

Random Day	Base	RS	2opt	MVLS
tempo	2610.0	1608.0	1517.0	1364.0
distanza	2615.0	1566.0	1461.0	1212.0
carico	0.22	0.18	0.18	0.41

Figura 4.2.1.2 Tabella riassuntiva per derivanti della schedulazione Random In Day

Greedy	Base	RS	2opt	MVLS
tempo	1432.0	1378.0	1315.0	1228.0
distanza	1304.0	1268.0	1222.0	1118.0
carico	0.3	0.3	0.33	0.47

Figura 4.2.1.3 Tabella riassuntiva per derivanti della schedulazione Greedy

In tutte le schedulazioni si può notare un effetto, più o meno accentuato, delle euristiche che a livello temporale e spaziale diminuiscono il costo delle soluzioni. Possiamo affermare che gli algoritmi applicati riescono a diminuire i costi.

L'andamento dei tempi va di pari passo con quello delle distanze, dato che in linea di massima la variazione di una comporta una variazione nella stessa direzione (aumento/diminuzione) dell'altra (figure 4.2.1.4 e 4.2.1.5). Questo effetto è dato dal fatto che se una tratta è più corta della seconda, presenterà un tempo minore, al contrario una tratta molto lunga avrà un tempo di percorrenza maggiore. Questa relazione è valida in entrambi i sensi, tempo->spazio e spazio->tempo. Di norma un tragitto con tempo di percorrenza basso risulterà più corto di uno con un tempo maggiore. Questa regola non vale per ogni possibile tragitto, dato che in una data strada si potrebbe percorrere una certa distanza in un tempo minore che in una strada con limiti di velocità più bassi. Dovendo visitare varie città vicine tra loro i veicoli in questione percorreranno strade simili (per limiti di velocità), per questo i due grafici non riportano esattamente le stesse variazioni, ma mantengono l'andamento generale. Poiché la differenza media tra il miglioramento percentuale del tempo ed il miglioramento percentuale dello spazio è solamente del 2,42%, è possibile affermare che tempo e spazio avranno sempre lo stesso andamento.

Non è fisso il miglioramento ottenibile applicando le varie euristiche: partendo da una soluzione pessima (random_inday) l'applicazione del reinsertion permette una diminuzione dei tempi del 38,3%, mentre per l'original ed il greedy la diminuzione è rispettivamente dell'11,7% e del 3,7%; le variazioni spaziali risultano simili: per il random_inday 40,11%, per l'original 13,36% e per il greedy 2,76%. In questi casi si può osservare che, partendo da una soluzione peggiore, il margine di miglioramento con l'applicazione del primo algoritmo (reinsertion) è maggiore rispetto a quello presentato dalle soluzioni generate da basi già buone. Con le casistiche presentate questo

fenomeno persiste in ogni applicazione delle euristiche, risulta tuttavia meno marcato nelle transizioni reinsertion-2opt e 2opt-MVLS.

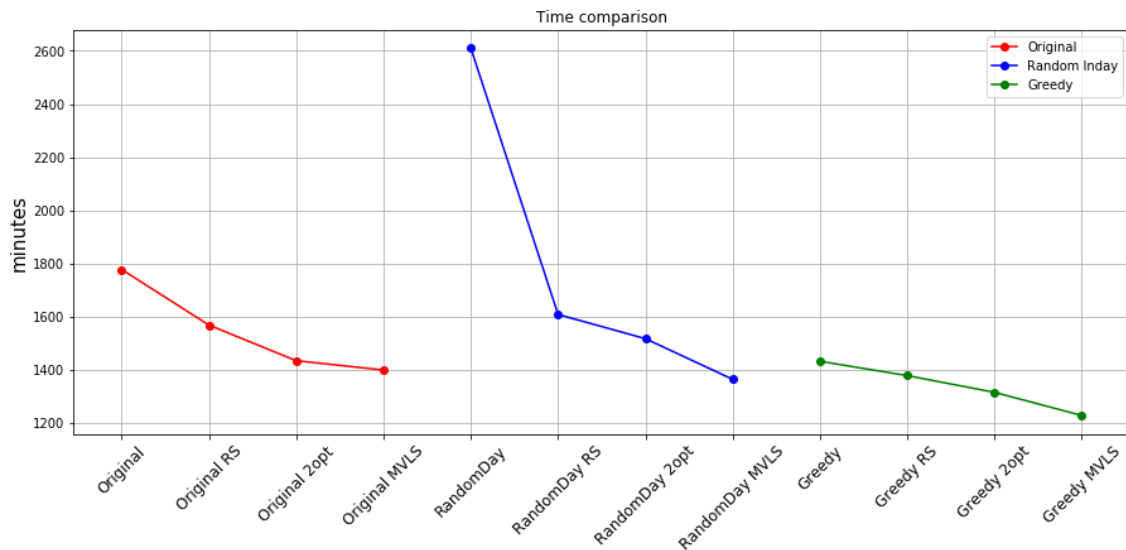


Figura 4.2.1.4 Grafico di comparazione dei tempi

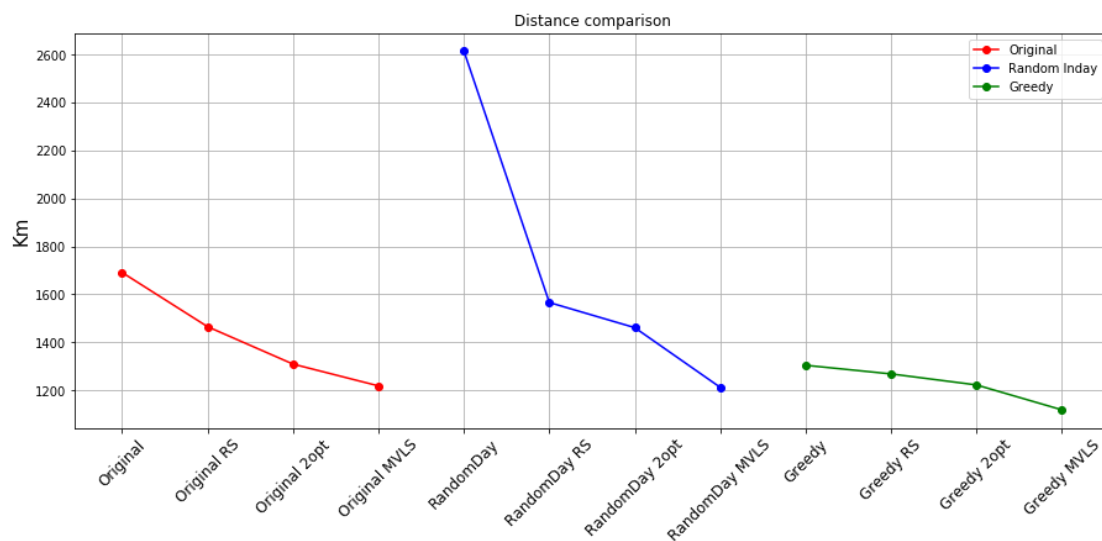


Figura 4.2.1.5 Grafico di comparazione delle distanze

L'andamento del carico non è detto coincida con quello dei tempi e delle distanze. Se si parte da una certa schedulazione con bilanciamento x , andando a migliorare i tempi per entrambi i veicoli il carico di lavoro potrebbe variare. Infatti, se si effettuano miglioramenti differenti (in termini di tempo/distanza) sui due veicoli si ottiene una soluzione che complessivamente presenta tempi e distanze minori, tuttavia risulta più sbilanciata. Un bilanciamento costante indica un taglio dei costi simile per entrambi i veicoli, una decrescita invece indica miglioramenti differenti che riportano i costi (temporali/spaziali) dei due veicoli allo stesso livello.

Per ogni transizione applicata con un algoritmo giornaliero si nota un miglioramento o mantenimento del bilanciamento. Con l'applicazione del MVLS l'indicatore del bilanciamento aumenta drasticamente in ogni caso esaminato (fino al 350% tra original_2opt e original_MVLS), questo perché si verifica un miglioramento del costo temporale che porta ad un accentramento delle sedi in un particolare giorno e veicolo, sbilanciando la soluzione.

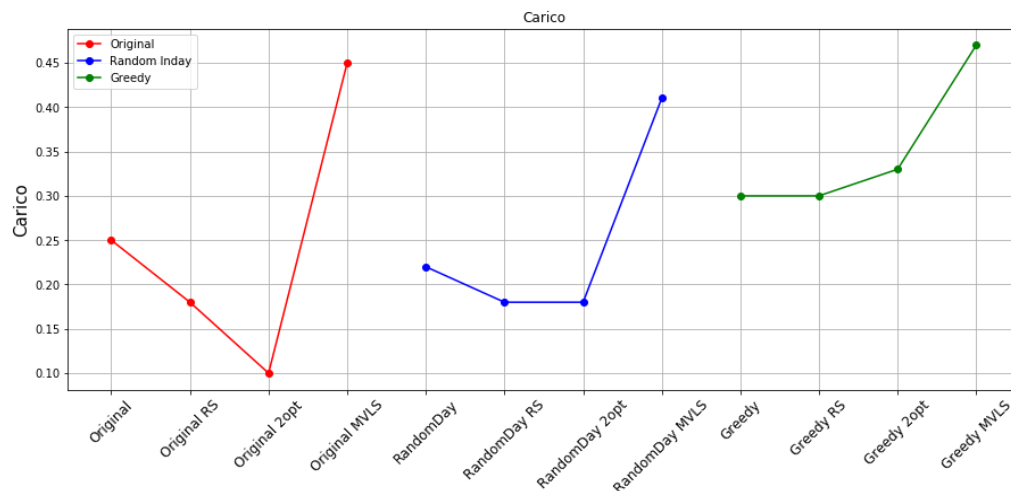


Figura 4.2.1.6 Grafico di comparazione del carico

Analizzati gli indicatori dei costi e del carico si procede al confronto dei KPI riguardanti il grado di servizio **mp** [4] ed **ms** [5] (figure 4.2.1.7, 4.2.1.8 e 4.2.1.9).

	original	original MVLS	random inday	random inday MVLS	greedy	greedy MVLS
mp	3.07	3.19	3.07	3.0	3.88	3.76
ms	4.51	4.21	4.51	4.34	4.22	4.2

Figura 4.2.1.7 tabella dei valori mp e ms

Dato che la distribuzione delle sedi nelle giornate rimane la stessa, i tempi di attesa per la schedulazione iniziale “originale” e per quella randomizzata giornalmente “random_inday” saranno uguali.

I tempi di attesa per la presa in carico presentano sia miglioramenti (2% nel caso del random_inday_MVLS rispetto al random_inday e 3% nel caso del greedy_MVLS rispetto al greedy) che peggioramenti (3% nel caso dell'original_MVLS rispetto all'original). Poiché gli incrementi e decrementi risultano molto piccoli, le variazioni

sono trascurabili. Da questi dati non è possibile affermare che il MVLS influisce positivamente o negativamente sull'indicatore **mp**.

Si nota un generale miglioramento del parametro **ms** nelle soluzioni generate dall'applicazione dell'algoritmo MVLS: una piccola diminuzione tra il greedy e il greedy_MVLS (dello 0.4%), una diminuzione intermedia tra random_inday e random_inday_MVLS (3.7%) ed infine una diminuzione più significativa tra l'original e l'original_MVLS (6.6%). Non si presenta una correlazione tra l'andamento degli indicatori **ms** e quelli **mp**, in alcuni casi all'aumento di uno corrisponde la diminuzione dell'altro ed in altri il contrario. Da uno dei due non è possibile ricavare informazioni sull'altro, dato che il tempo medio per la presa in carico (**mp**) concorre a formare il tempo medio per il completamento di un ordine (**ms**) insieme ad un indicatore derivabile dalla differenza dei due, il tempo medio per la consegna dal momento della presa in carico. Una variazione di **mp** in un senso e di **ms** nell'altro significherebbe solamente che il tempo medio per il compimento di un ordine (dal giorno di presa in carico alla consegna) è aumentato o diminuito. Da uno dei due non è possibile avere informazioni sull'altro.

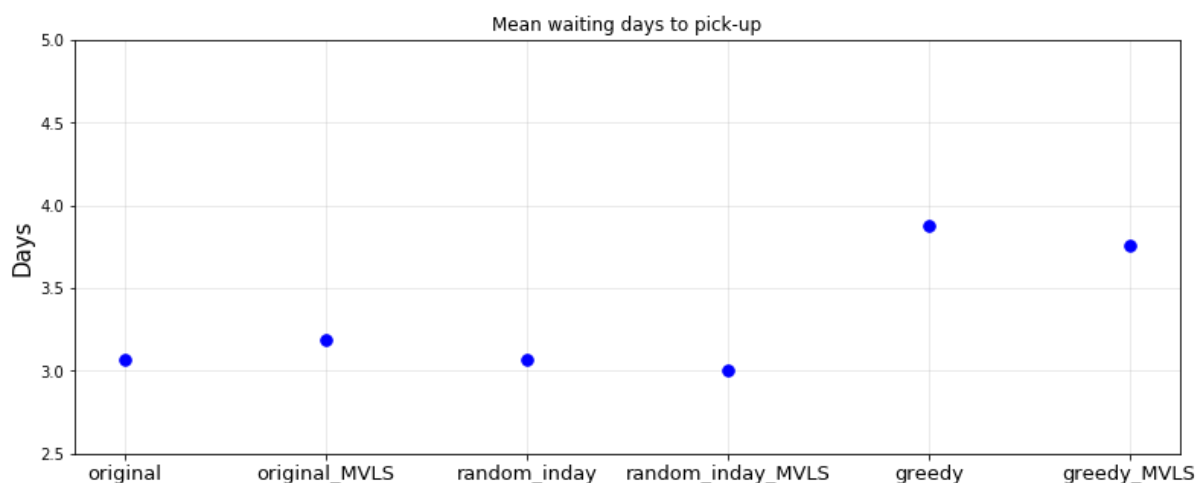


Figura 4.2.1.8 Tempi medi di attesa per la presa in carico di un ordine

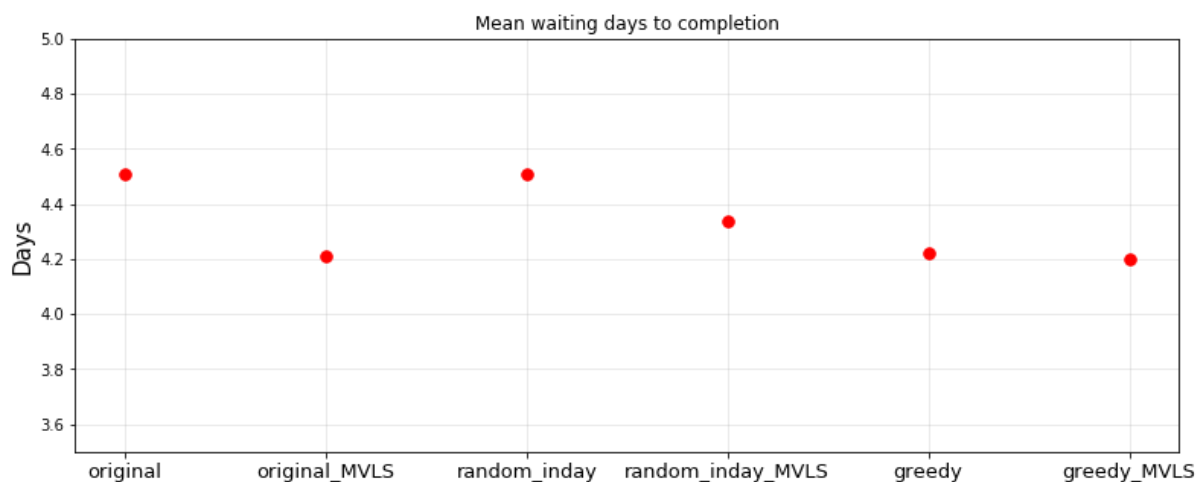


Figura 4.2.1.9 Tempi medi di attesa per il completamento di un ordine

4.2.2 - Schedulazioni con base calcolata

In questa sezione saranno confrontate le soluzioni generate basandosi sui risultati dei modelli OSS, FOSS e sulla schedulazione randomica. In riferimento alla figura 3.2.1 saranno presentate le soluzioni derivanti da Gurobi e Random Schedule. Queste soluzioni modificano l'assegnamento delle sedi alle giornate lavorative sin dalla partenza, disponendole in modo tale da migliorare il grado di servizio (OSS e FOSS) o casualmente.

Su ognuna di esse verranno applicati, come sulle soluzioni derivanti dalla schedulazione originale, le euristiche in sequenza (suddivisione greedy, reinsertion, 2opt e MVLS).

Di seguito sono riportate le tabelle riassuntive delle schedulazioni derivanti dalla base OSS (figura 4.2.2.1), dalla base FOSS (figura 4.2.2.2) e dalla base casuale Random Schedule (figura 4.2.2.3). Per ognuna di esse sono riportati i KPI riguardanti **tempo** [1], **distanza** [2] e **carico** [3]. Si vorrà verificare se un'analisi preliminare sia efficace per migliorare il grado di servizio, disponendo al meglio le sedi nelle varie giornate, e se le euristiche applicate non peggiorino la qualità ottenuta tramite l'utilizzo dei modelli.

OSS	Greedy	RS	2opt	MVLS
tempo	1687.0	1636.0	1475.0	1426.0
distanza	1612.0	1555.0	1341.0	1289.0
carico	0.25	0.24	0.19	0.46

Figura 4.2.2.1 Tabella riassuntiva per derivanti della schedulazione OSS

FOSS	Greedy	RS	2opt	MVLS
tempo	1602.0	1582.0	1443.0	1392.0
distanza	1485.0	1477.0	1328.0	1247.0
carico	0.37	0.37	0.31	0.48

Figura 4.2.2.2 Tabella riassuntiva per derivanti della schedulazione FOSS

Random Schedule	Greedy	RS	2opt	MVLS
tempo	1869.0	1746.0	1643.0	1419.0
distanza	1798.0	1685.0	1545.0	1262.0
carico	0.17	0.22	0.22	0.35

Figura 4.2.2.3 Tabella riassuntiva per derivanti della schedulazione Random_Schedule

Come nei casi trattati nella sezione 4.2.1 si nota un andamento decrescente per i parametri temporale e spaziale. Per tutte e tre le basi di partenza questi indici variano di pari passo; la forma dei grafici risulta molto simile per ogni andamento tempo-spazio della stessa schedulazione (figure 4.2.2.4 e 4.2.2.5). Anche in questo caso si vedono piccole discrepanze tra l'andamento temporale e spaziale dovuti alle limitazioni di velocità sulle strade percorse che limitano lo spazio percorribile in un dato tempo; random schedule, nel passaggio tra 2opt e MVLS registra una diminuzione del 13,63% a livello temporale e del 18,32% a livello spaziale, con una discrepanza del 4,68%. La differenza media tra miglioramento temporale e spaziale risulta dell'1,88%; queste differenze sono tuttavia minime e trascurabili, modifiche su tempo o spazio influenzano nella stessa direzione anche l'altro indicatore, così come era stato confermato dai valori delle schedulazioni nella sezione 4.2.1. Come nei casi precedentemente presentati, reinsertion e 2opt non influiscono negativamente sul bilanciamento del carico nelle prime due schedulazioni mentre lo peggiorano nella terza; per questi algoritmi non è possibile prevedere l'effetto sul carico dato dalla loro applicazione; il MVLS invece permette sempre un miglioramento delle tempistiche e del chilometraggio, tuttavia in tutti i casi esaminati il carico risulta sbilanciato.

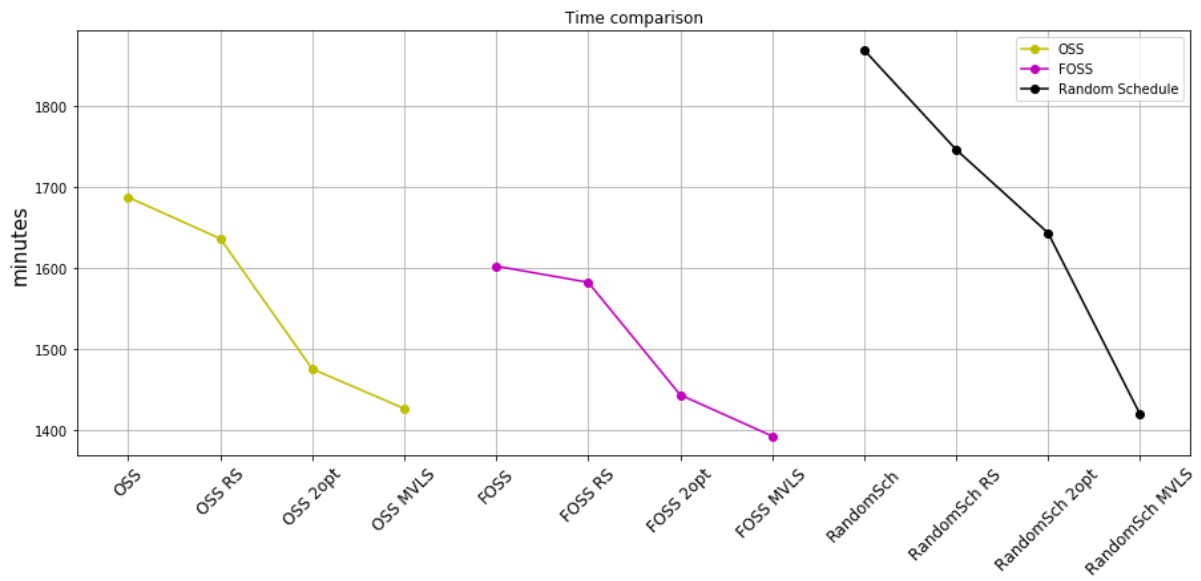


Figura 4.2.2.4 Grafico di comparazione dei tempi

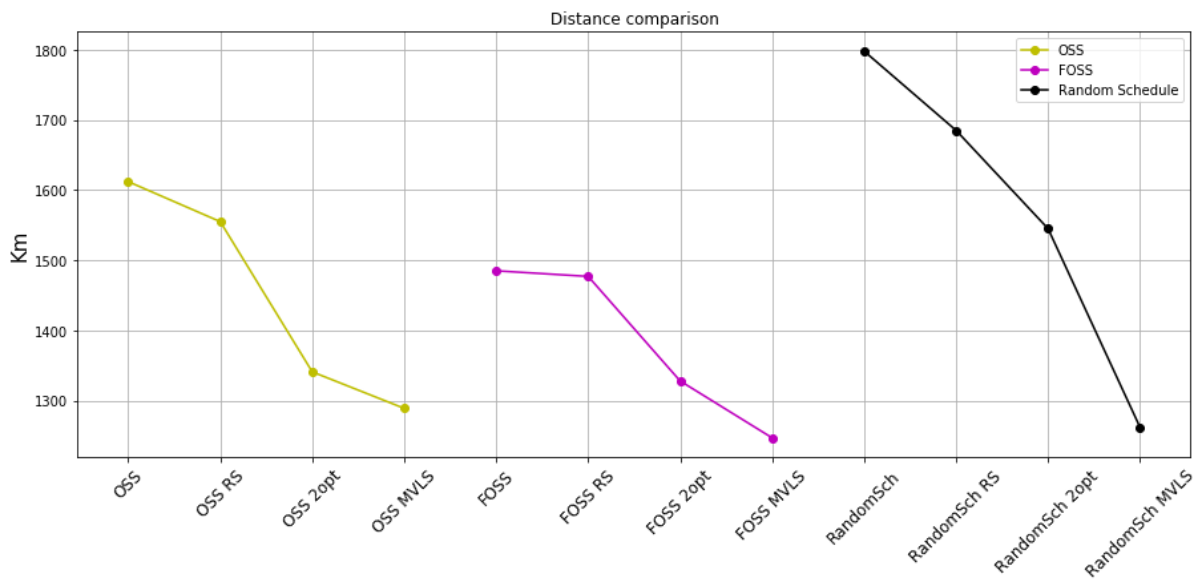


Figura 4.2.2.5 Grafico di comparazione delle distanze

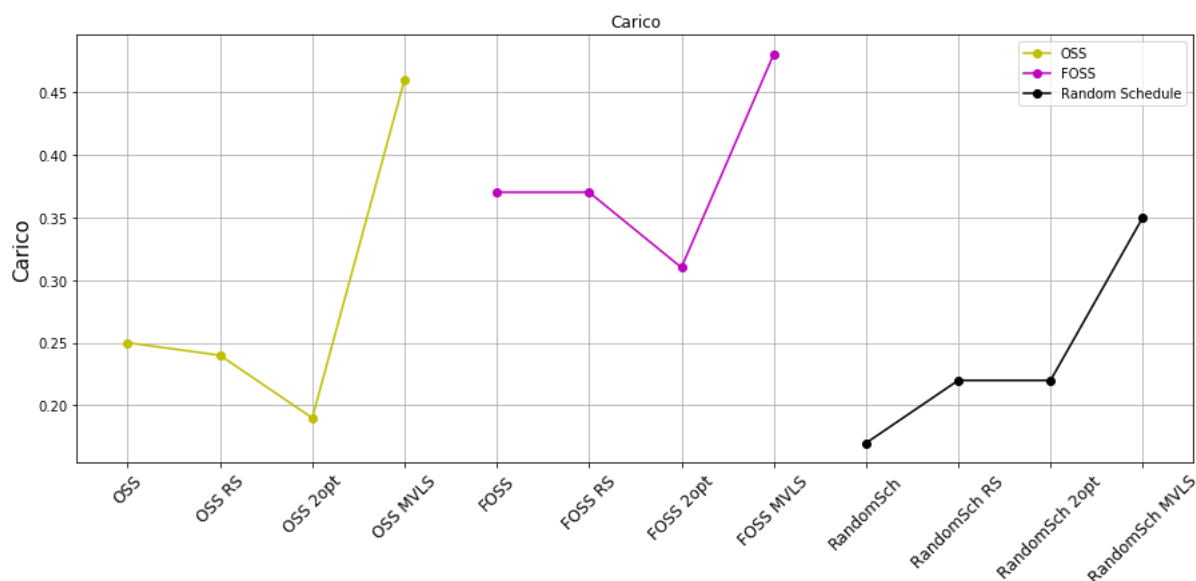


Figura 4.2.2.6 Grafico di comparazione del carico

Poiché OSS e FOSS sono schedulazioni generate appositamente per ottimizzare il grado di servizio risulta importante confrontarne **mp** ed **ms** per verificarne l'efficacia. A priori si può ipotizzare l'indice **ms** migliore nella schedulazione FOSS, essendo costruita appositamente per minimizzare questo valore, seguito da OSS, modello che ottimizza meno finemente questo valore. Non è possibile avanzare ipotesi riguardo al random_schedule, dato che la sua composizione è generata casualmente. Riguardo ai valori di **mp** non è possibile prevedere il comportamento nelle varie schedulazioni, non essendo ottimizzate per questo indicatore.

	OSS	OSS MVLS	FOSS	FOSS MVLS	random schedule	random schedule MVLS
mp	3.22	4.08	3.06	3.07	3.25	3.85
ms	4.3	4.32	4.09	4.42	4.12	4.29

Figura 4.2.2.7 Grafico di comparazione dei tempi

In riferimento alla tabella sopra riportata (figura 4.2.2.7) i valori di **mp** rispecchiano le ipotesi avanzate per i valori di **ms**: pur non essendo ottimizzati per migliorare il tempo medio per la presa in carico OSS e FOSS presentano i valori migliori, rispettivamente 3.22 e 3.06. La schedulazione randomica presenta un valore (3,25) molto simile a quello dell'ordinamento OSS. In tutti i casi l'applicazione del MVLS peggiora le tempistiche; il caso peggiore di applicazione risulta l'OSS, dato che si ha un peggioramento del 26,7%,

il caso migliore rimane il FOSS, che porta un peggioramento del solo 0,3%. Nonostante i due modelli non siano stati creati per ottimizzare questo KPI, i valori di **mp** seguono la tendenza che ci si sarebbe aspettati da **ms**.

Riguardo al tempo medio di attesa per il completamento dell'ordine, FOSS risulta il migliore seguito però dalla schedulazione randomica che presenta un valore **ms** molto simile. L'ordinamento ottenuto tramite modello OSS risulta il peggiore tra i modelli base.

Per quanto riguarda l'applicazione del MVLS in tutti i casi viene rilevato un aumento dei tempi di attesa, il caso peggiore risulta quello del FOSS_MVLS con un incremento dell'8,06% rispetto all'**ms** della schedulazione di base.

Seppur di poco, l'applicazione di un modello (FOSS) destinato all'ottimizzazione del tempo di attesa per il completamento degli ordini risulta la scelta migliore in base al KPI **ms**. Va ricordato che anche se la schedulazione casuale ha riportato questi risultati, non è assicurabile ottenere KPI ottimi con una scelta randomica della schedulazione. Random_schedule è servita per provare la miglior ottimizzazione, almeno per il modello FOSS, riguardo l'indicatore **ms**.

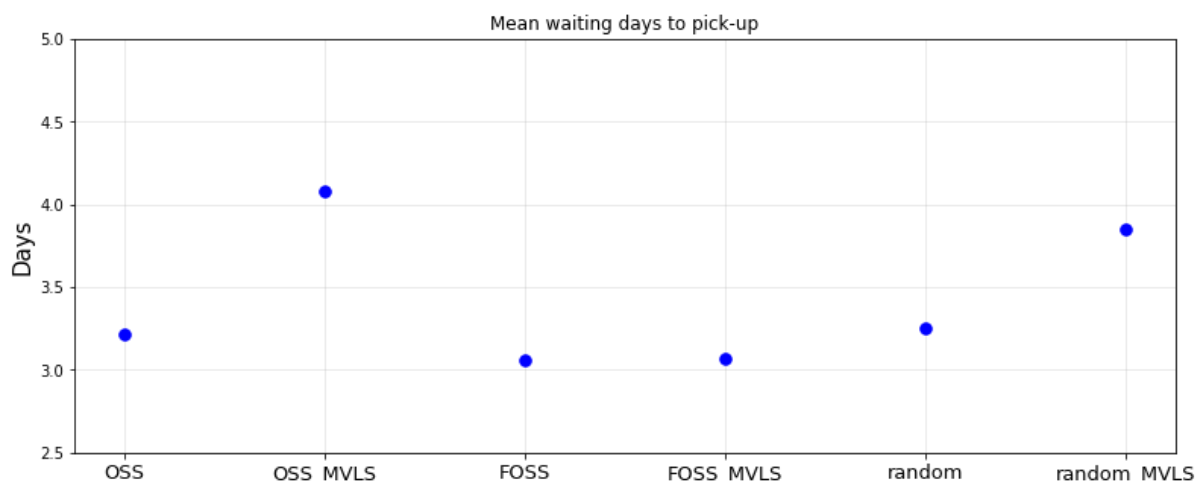


Figura 4.2.2.8 Tempi medi di attesa per la presa in carico di un ordine

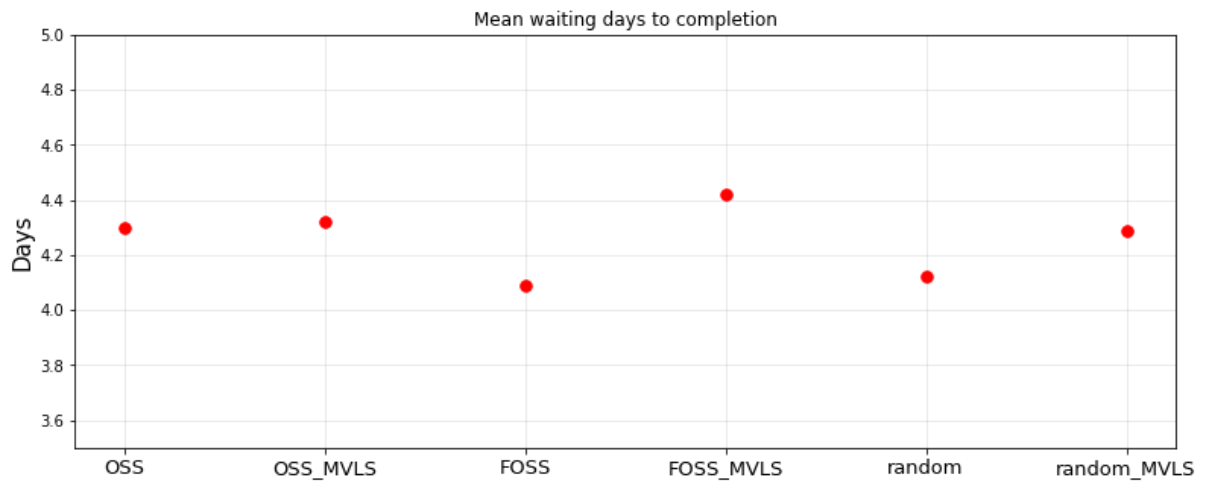


Figura 4.2.2.9 Tempi medi di attesa per la presa in carico di un ordine

4.2.3 - Confronto totale

In questa sezione verranno confrontati tutti i risultati ottenuti, cercando di sottolineare pregi e difetti ed evidenziando eventuali possibili soluzioni al problema della rete bibliotecaria cremonese. Per fare ciò bisogna tenere conto di tutti i KPI e valutare vantaggi e svantaggi di ogni soluzione.

A livello di sole tempistiche e chilometraggio le soluzioni migliori risultano essere il greedy_MVLS, seguito dal random_inday_MVLS e dal FOSS_MVLS (figure 4.2.1.3, 4.2.1.2 e 4.2.2.2), le quali tuttavia sono probabilmente impraticabili a causa dello sbilanciamento del carico: in generale l'applicazione dell'algoritmo multi vehicle local search rende le soluzioni sbilanciate, per questo è necessario prendere in considerazione soluzioni meno performanti, ma più equilibrate.

Escludendo i casi del greedy e random_schedule, l'algoritmo 2opt riduce sia i costi (spazio-temporali) che il carico (figura 4.2.3.3). L'algoritmo di reinsertion presenta i medesimi effetti, tuttavia in maniera più blanda: i miglioramenti del reinsertion non arrivano al livello di quelli del 2opt. Dunque l'applicazione in sequenza base-reinsertion-2opt si presenta come la miglior via risolutiva.

Tra le soluzioni 2opt le migliori, a livello di costi, sono greedy_2opt, original_2opt e FOSS_2opt: il primo mostra uno sbilanciamento che rende impraticabile la soluzione; gli altri due, nonostante siano simili a livello di tempo e chilometraggio (figura 4.2.3.1), si differenziano a livello di carico (in quanto l'original risulta migliore) e a livello di tempo di attesa per il completamento dell'ordine (dove risulta più proficuo il secondo).

Considerando il solo bilanciamento del carico sarebbe opportuno scegliere l'original_2opt, che tuttavia introduce un tempo di attesa per il completamento degli ordini (**ms**) maggiore del 31,9% rispetto al tempo di attesa medio della soluzione FOSS.

In conclusione, volendo mantenere la schedulazione più bilanciata possibile, a discapito del grado di servizio, la soluzione più indicata è original_2opt; privilegiando la soluzione con tempo medio di attesa si deve optare per il FOSS_2opt.

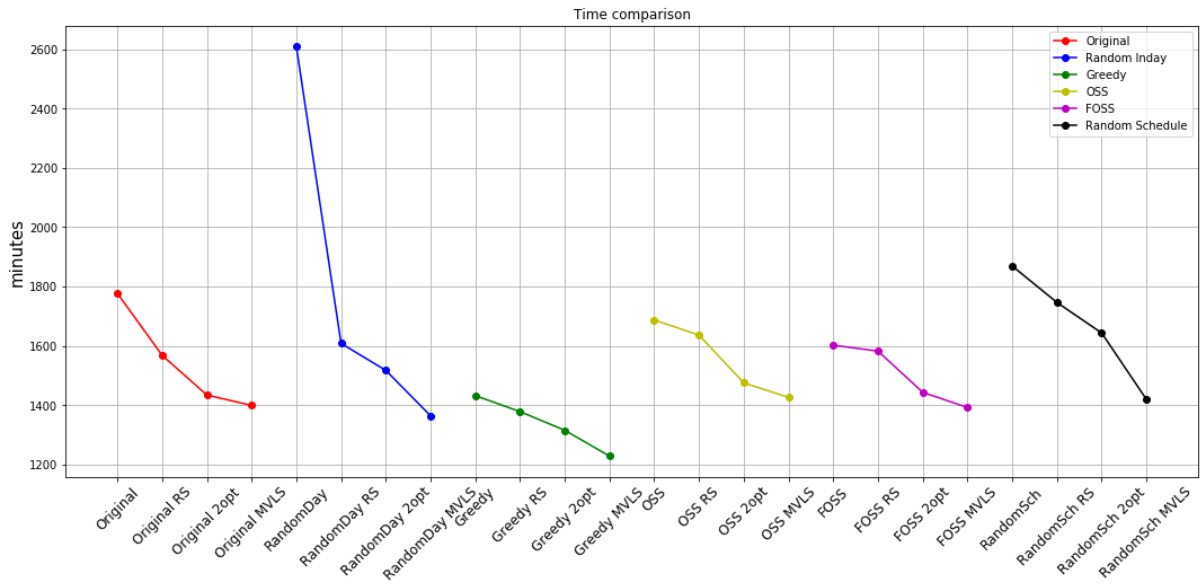


Figura 4.2.3.1 Grafico comparativo temporale di tutte le schedulazioni

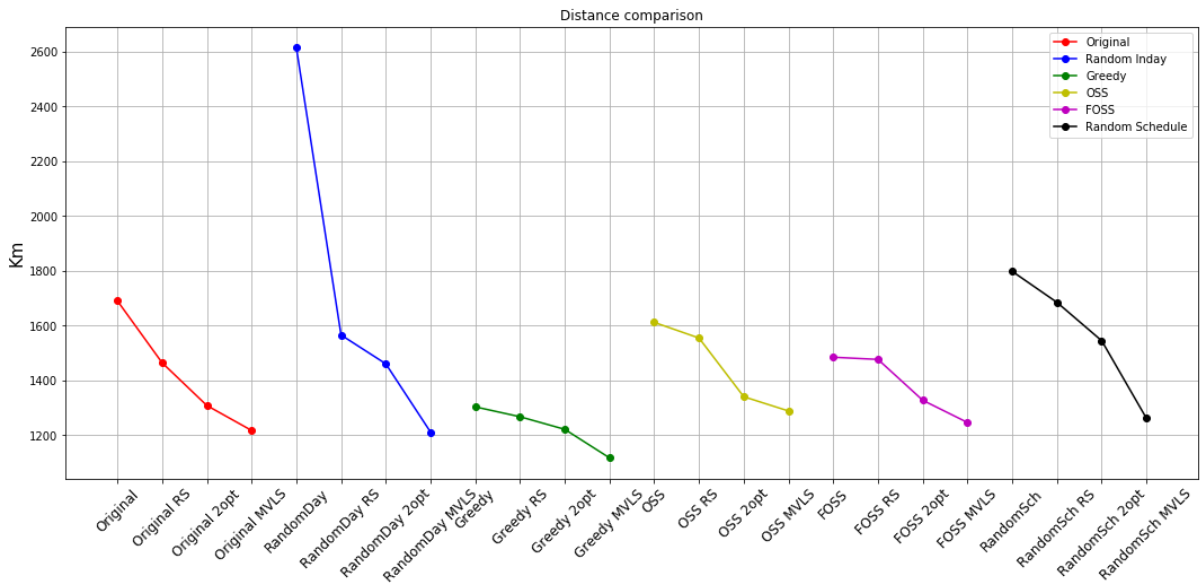


Figura 4.2.3.2 Grafico comparativo spaziale di tutte le schedulazioni

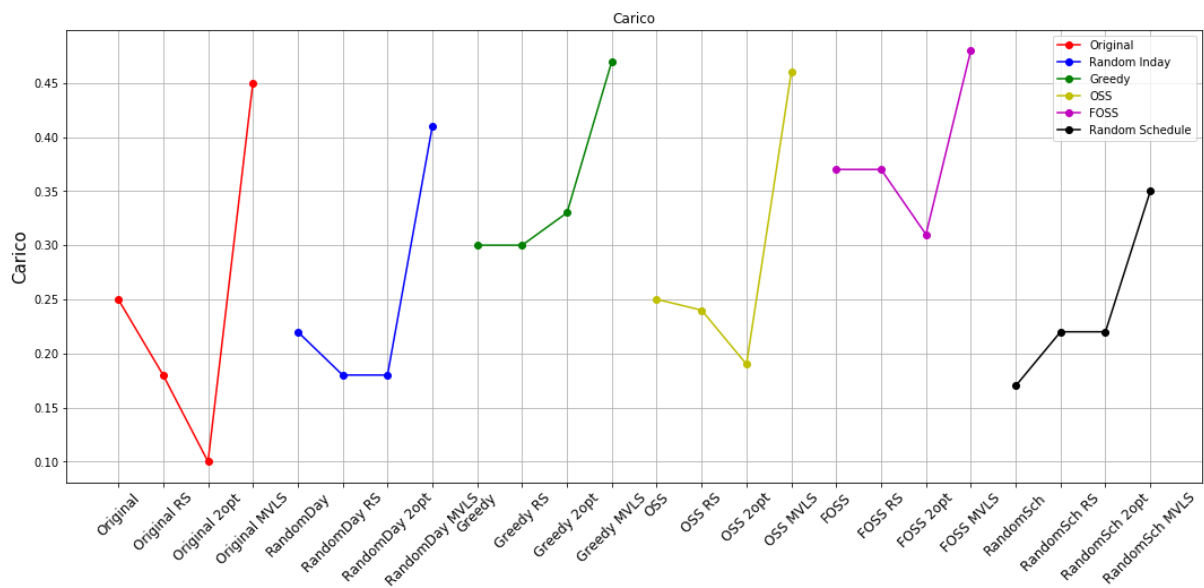


Figura 4.2.3.3 Grafico comparativo del bilanciamento del carico di tutte le schedulazioni

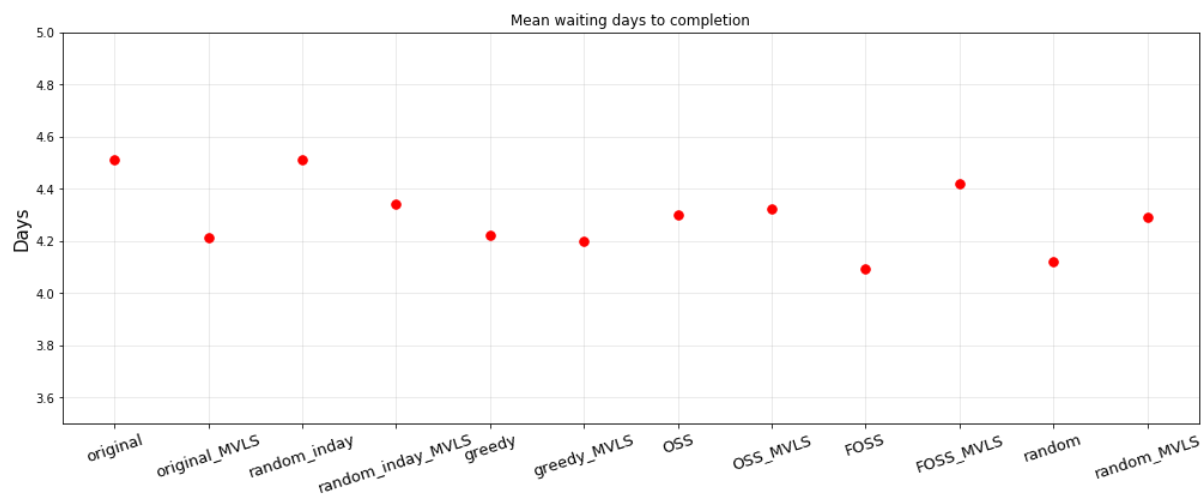


Figura 4.2.3.4 Grafico comparativo di giorni medi di attesa per il completamento dell'ordine

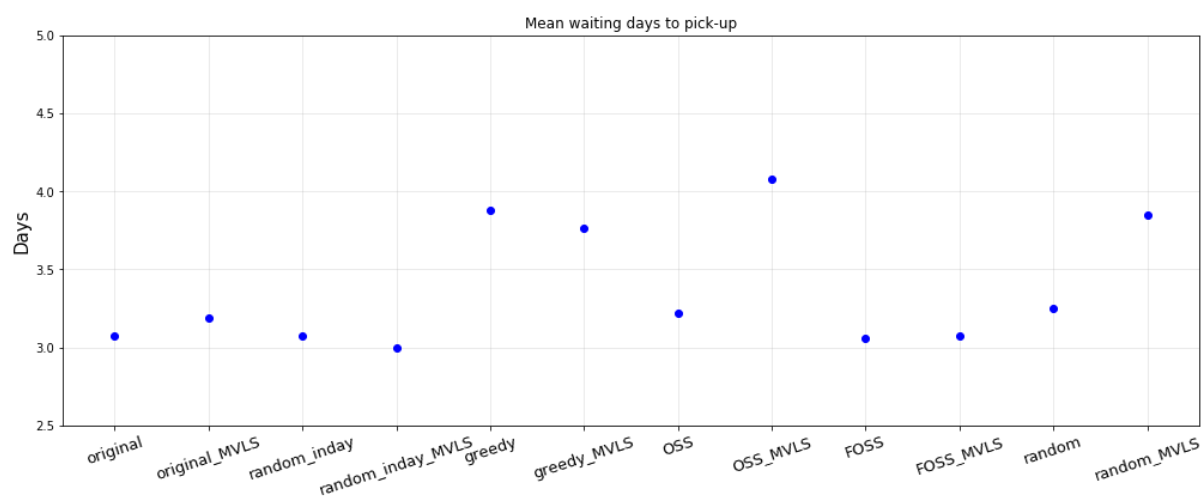


Figura 4.2.3.5 Grafico comparativo di giorni medi di attesa per la presa in carico dell'ordine

Capitolo V: Conclusioni

Nell'elaborato, per poter affrontare appieno il problema, è stata effettuata un'analisi dei dati forniti per comprenderne appieno la natura ed estrapolarne potenziali informazioni utili, non direttamente interpretabili. L'analisi preliminare ha permesso inoltre di isolare quali aspetti sono importanti per la valutazione delle soluzioni tramite la creazione di appositi indicatori (KPI), da questi sono stati fatti emergere quali sarebbero stati i punti cruciali da migliorare:

[1] tempistiche/chilometraggio;

[2] grado di servizio.

L'identificazione degli aspetti più importanti da ottimizzare ha permesso di selezionare in quali modi migliorare ognuno.

Gli aspetti tempo-spaziali sono stati migliorati tramite l'applicazione di algoritmi euristici[1]; l'ottimizzazione del grado di servizio risulta più complessa ed è stata affrontata tramite l'uso di modelli di programmazione (OSS e FOSS) [2].

L'impiego di entrambe le metodologie sopra citate ha permesso di generare una base che cerca di stimare il miglior grado di servizio (a diversi livelli di finezza), per poi applicare le euristiche in modo da ottimizzare il costo (tempo/spazio).

I KPI costruiti hanno permesso di valutare tutte le possibili soluzioni generate, sotto tutti gli aspetti di interesse per confrontarli al fine di selezionare possibili schedulazioni attuabili dalla rete bibliotecaria cremonese.

La valutazione delle potenziali soluzioni ha portato alle seguenti conclusioni:

- L'impiego dei modelli di ottimizzazione restituisce soluzioni migliori a livello di servizio, senza andare a discapito dei costi tempo-spaziali;
- Le euristiche operanti nella singola giornata risultano buone per migliorare gli indicatori di performance progettati;
- L'euristica operante intergiornata riesce a diminuire i costi temporali e spaziali, ma spesso rende la soluzione molto sbilanciata e quindi potenzialmente inammissibile.

Bibliografia - Sitografia

Ultimo accesso dicembre 2020

- (1) Servizi interbibliotecari: [wikipedia](https://www.wikipedia.org/)
- (2) Data analysis: [reflectivedata.com](https://www.reflectivedata.com/)
- (3) Miglioramenti aziendali con applicazione di data analysis: [newvantage.com](https://www.newvantage.com/)
- (4) Esempi data analysis: [processgold.com](https://www.processgold.com/)
- (5) Definizione modello matematico: M. Caramia, S. Giordani, F. Guerriero, R. Musmanno, D. Pacciarelli - *Ricerca operativa* (2014), Capitolo I
- (6) Definizione vehicle routing: Toth P. and Vigo D., editors. *The Vehicle Routing Problem. SIAM monograph on discrete mathematics and applications* (2014).
- (7) Definizione analisi: Sheldon M. Ross - *Probabilità e statistica per l'ingegneria e le scienze* (2003, Apogeo)
- (8) Definizione analisi descrittiva: Sheldon M. Ross - *Probabilità e statistica per l'ingegneria e le scienze* (2003, Apogeo), Capitolo II
- (9) Definizione modello di programmazione: M. Caramia, S. Giordani, F. Guerriero, R. Musmanno, D. Pacciarelli - *Ricerca operativa* (2014), Capitolo I
- (10) Definizione algoritmo euristico: Jon Kleinberg, Éva Tardos - *Algorithm Design*, (Pearson 2006)
- (11) Definizione local search: Jon Kleinberg, Éva Tardos - *Algorithm Design*, (Pearson 2006), Capitolo 12
- (12) Definizione greedy: Jon Kleinberg, Éva Tardos - *Algorithm Design*, (Pearson 2006), Capitolo 4

Stesura del codice

- (13) Fonte Anaconda: [anaconda.com](https://www.anaconda.com/)
 - (14) Fonte Jupyter: jupyter.org
 - (15) Fonte GitHub: github.com
 - (16) Fonte Gurobi: [gurobi.com](https://www.gurobi.com/)
- Fonte Stackoverflow: stackoverflow.com
- Repository: github.com/Bertonc98/Tesi

Ringraziamenti

Vorrei ringraziare il Professor Alberto Ceselli che ha reso possibile questo progetto e che mi ha fatto crescere in questo ambito di ricerca grazie alle sue conoscenze e disponibilità.

I miei amici di sempre Camilla, Claudia, Claudio, Davide, Gabriele, Jacopo, Lorenzo, Luca Ma., Luca Mo., Luca Ri., Luca Ro., Mattia, Marco, Nicole, Simone, Vanessa persone insostituibili, confidenti e consiglieri per i problemi che mi trovo ad affrontare.

I miei colleghi, in particolare Massimo compagno di tirocinio che mi ha sopportato e soprattutto supportato in questi mesi da vero amico; *Alessandro*, Davide, Elan, Elia, Francesca A., Francesca F., Giorgio, Giulia, Marco Ce., Marco Ca., Nicolas e Riccardo che mi sono stati affianco durante il viaggio stupendo vissuto in questi anni.

Gli zii Giuseppe e Gianpiera che nonostante il cupo periodo mi hanno spronato a fare del mio meglio oltre ad essere sempre disponibili e generosi.

Beatrice che mi ha sempre sostenuto, sia nei momenti più felici esultando con me che in quelli in cui ero sul punto di mollare, credendo in me e dandomi la forza di continuare; Tony, Katia, Gabriele e tutta la loro famiglia che ormai sono una seconda casa e rappresentano un posto in cui mi sento sempre ben accolto e felice.

I miei genitori, Umberto e Carla, per le possibilità che mi hanno sempre garantito e per il sostegno dato in tutti questi anni; ci sono state difficoltà, ma hanno sempre cercato di incoraggiarmi sul percorso che ho scelto.