

Università degli Studi dell'Insubria -Varese
Dipartimento di Informatica
Corso di laurea triennale in Informatica



OTTIMIZZAZIONE DEI TRASPORTI

*Analisi sull'uso di incentivi nel food delivery per la
creazione di un percorso ultimo ottimizzato.*

Tesi di:
Andrea Morandi

Relatore:
Prof. Ignazio Gallo

Matricola: 732923
Anno Accademico 2019/2020

Indice

Introduzione	4
1 Analisi letteratura	5
1.1 Vehicle Routing Problem e Vehicle Routing Problem with Time Windows.....	5
1.2 Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problem	6
1.3 Home Delivery Problem with Time Slot Incentives	7
1.4 Foresight Policy	8
1.5 Problema dell'azienda	9
2 Strumenti utilizzati	10
2.1 Introduzione JavaScript	10
2.2 Linguaggio di scripting e interpretato	10
2.3 Linguaggio orientato agli oggetti	10
2.4 JSON.....	11
2.5 Motivazione della scelta.....	11
2.6 API.....	11
2.7 API scelte per il progetto.....	13
3 Progetto tesi	14
3.1 Introduzione.....	14
3.2 Clienti	14
3.3 Fasce orarie.....	14
3.4 Incentivi.....	15
3.5 Valutazione fattibilità fascia oraria e relativo costo: algoritmo euristico.....	15
3.6 Valutazione fattibilità fascia oraria e relativo costo: API per percorsi ottimizzati	16
3.7 Previsione nella valutazione del costo di inserimento del nuovo ordine: Foresight Policy	16
3.8 Previsione nella valutazione del costo di inserimento del nuovo ordine: peso fasce orarie ...	17
4 Analisi simulazioni	22
4.1 Simulazioni.....	22
4.2 Intuizioni e analisi della politica di offerta delle fasce orarie	28
4.3 Conclusioni e futuri sviluppi	29
5 Bibliografia e sitografia	31

Abstract

Molte aziende di servizi rivolti al cliente, specialmente per quanto riguarda la consegna di generi alimentari (food delivery), devono affrontare il “problema dell’ultimo miglio”, ovvero la consegna effettiva dell’oggetto al cliente presso il domicilio o il negozio. Questo problema logistico è reso estremamente complesso da fattori quali l’imprevedibilità della domanda e i limiti imposti dalle fasce orarie. In questa tesi si analizza l’uso di incentivi/disincentivi per influenzare la scelta dei consumatori verso le fasce orarie ritenute adatte per creare un percorso ultimo di consegne ottimizzato. A tal fine viene inoltre proposta una politica di offerta delle fasce orarie basata sul percorso parziale ottenuto dagli ordini già accettati.

Introduzione

Negli ultimi anni numerose aziende stanno affrontando la sfida della digitalizzazione, offrendo ai clienti la possibilità di acquistare online e ottenere la spedizione direttamente a casa o in negozio. Questo servizio può essere suddiviso in tre fasi: (1) emissione dell’ordine, (2) approvvigionamento e assemblaggio dell’ordine, (3) consegna dell’ordine. La tesi discute il rapporto tra la prima e l’ultima fase, analizzando l’impatto dell’uso di incentivi per influenzare la scelta del consumatore verso alcune fasce orarie, ritenute le più convenienti per l’azienda. Il lavoro coinvolge due aspetti principali: il problema dell’ottimizzazione dei trasporti e la gestione dei ricavi. Il primo è un problema logistico, il cui scopo è quello di creare il più breve itinerario possibile che rispetti i vincoli dati (fasce orarie, quantità, ...). Per gestione dei ricavi si intende invece “il sistema di gestione delle capacità disponibili che ha come obiettivo la massimizzazione e l’ottimizzazione del volume di affari” [6]. Lo scopo ultimo è quindi quello di sfruttare al meglio le risorse disponibili (veicoli, personale, ...) per accettare il maggior numero possibile di ordini nell’arco della giornata lavorativa.

Per studiare i punti presentati è stata creata una pagina web che permette di simulare le scelte dei clienti a seguito dell’offerta di determinate fasce orarie con relativi incentivi e disincentivi. La tesi è stata realizzata come progetto interno all’Università e in collaborazione con l’azienda “Planeat”, la quale si occupa di consegne di prodotti alimentari a domicilio, attualmente nelle zone di Pavia e Milano. Il lavoro è organizzato come segue. Nel primo capitolo si analizzano il problema e la letteratura esistente e si conclude introducendo la situazione dell’azienda. Nel secondo si presentano gli strumenti utilizzati nello svolgimento del progetto, ovvero linguaggi e API. Nel terzo viene descritta la soluzione ideata. Nel quarto e ultimo capitolo si valuta la soluzione tramite i risultati delle simulazioni.

1 Analisi letteratura

In questo capitolo viene descritto il problema dell'ottimizzazione del trasporto (sezione 1.1), presentate le ricerche analizzate (sezioni 1.2, 1.3 e 1.4) e si conclude introducendo il problema specifico dell'azienda (sezione 1.5).

1.1 Vehicle Routing Problem e Vehicle Routing Problem with Time Windows

Il problema dell'ottimizzazione del trasporto, ovvero il Vehicle Routing Problem (VRP), può essere considerato come generalizzazione del Travelling Salesman Problem (TSP), nel quale “si richiede la determinazione di un circuito hamiltoniano a costo complessivo minimo” [2]. La descrizione più famosa di quest'ultimo problema è la rappresentazione in cui un commesso viaggiatore deve visitare una e una sola volta un insieme di città percorrendo la minima distanza possibile. Il VRP ne estende le caratteristiche e “comporta la ricerca di un insieme di rotte per i veicoli di una flotta in modo che ciascun veicolo parta dal deposito e vi ritorni al termine della rotta associata e che allo stesso tempo un dato insieme di clienti sia interamente servito. [...] Ciascun cliente ha una domanda specifica che deve essere soddisfatta dalla consegna e nessun veicolo può servire più clienti di quanto gli permetta di fare la propria capacità. L'obiettivo può essere quello di minimizzare la distanza totale percorsa dalla flotta o il numero di veicoli usati o una combinazione di questi obiettivi” [1].

Gli elementi primari del problema sono quindi la rete stradale e i clienti. “La rete stradale utilizzata per il trasporto normalmente viene rappresentata mediante un grafo i cui archi rappresentano i tratti stradali percorribili ed i cui vertici corrispondono ai punti notevoli della rete, cioè gli incroci ed ai punti ove sono localizzati clienti e depositi” [2]. Ogni cliente è invece caratterizzato dalla quantità di merce ordinata, intervalli di tempo o finestre temporali in cui può essere raggiunto ed “eventuale sottoinsieme di veicoli che possono essere utilizzati per servirlo (ad esempio, per i clienti di un centro storico possono essere adatti solo veicoli di piccola dimensione)” [2]. Il problema presenta numerose varianti a seconda dei vincoli imposti (capacità disponibili, fasce orarie, ...). “La verifica dei vincoli operativi richiede la conoscenza della distanza o del tempo di viaggio che separa una qualunque coppia di clienti o i depositi da un qualsiasi cliente. A tal fine normalmente viene definito, a partire dalla rete stradale, un grafo completo i cui vertici sono i vertici della rete corrispondenti a clienti e depositi. Per ciascuna coppia di vertici i e j del grafo esiste un arco il cui costo c_{ij} è il costo del percorso minimo, in termini di distanza o tempo di percorrenza, tra i due vertici, misurato sulla rete stradale originaria” [2]. Nello svolgimento della tesi ci si è concentrati

sul Vehicle Routing Problem with Time Windows (VRPTW), in cui il vincolo è dato dalla presenza delle finestre temporali.

1.2 Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems

La pubblicazione di Campbell e Savelsbergh [2004] introduce una metodologia per costruire una soluzione fattibile, ovvero dei tragitti che visitino un insieme di clienti, che rispetti i vincoli imposti dalle diverse varianti del VRP. L'idea è quella di valutare l'inserimento di ogni ordine j , non ancora compreso in una rotta, tra ogni altra coppia di ordini i e $i-1$ già accettati. Per ogni posizione si valuta la fattibilità dell'inserimento, dipendente dai vincoli del problema, e il relativo profitto. Al termine dell'iterazione si inserisce l'ordine con il maggior profitto. L'algoritmo viene descritto nella Figura 1.

Algorithm 1 Insertion Heuristic

```

1.  $N$  = set of unassigned customers
2.  $R$  = set of routes; always contains the empty route; initially contains
   only the empty route
3. while  $N \neq \emptyset$  do
4.    $p^* = -\infty$ 
5.   for  $j \in N$  do
6.     for  $r \in R$  do
7.       for  $(i-1, i) \in r$  do
8.         if  $Feasible(i, j)$  and  $Profit(i, j) > p^*$  then
9.            $r^* = r$ 
10.           $i^* = i$ 
11.           $j^* = j$ 
12.           $p^* = Profit(i, j)$ 
13.        end if
14.      end for
15.    end for
16.  end for
17.   $Insert(i^*, j^*)$ 
18.   $N = N \setminus j^*$ 
19.   $Update(r^*)$ 
20. end while

```

Figura 1: algoritmo di inserimento euristico. Campbell, Ann & Savelsbergh, Martin. (2004). Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. Transportation Science. 38. 369-378. 10.1287/trsc.1030.0046, p. 5.

La versione più rilevante nello svolgimento dell'elaborato è quella relativa al VRPTW, in cui i vincoli derivano dalle fasce orarie. In particolare si determinano il primo e ultimo istante in cui può

essere effettuata la consegna del nuovo ordine j , da inserire tra due ordini $i-1$ e i , con e e l ad indicare i primi momenti per servire il cliente, E e L ad indicare i limiti imposti dalla fascia oraria e T ad indicare i tempi di percorrenza [3]:

$$e_j = \max (E_j, e_{i-1} + T_{i-1,j}),$$

$$l_j = \min(L_j, l_i - T_{j,i}).$$

L'inserimento può essere accettato se:

$$e_j \leq l_j.$$

I profitti vengono calcolati in base alle distanze:

$$- (T_{i-1,j} + T_{j,i} - T_{i-1,i}) .$$

L'inserimento del nuovo ordine comporta la necessità di aggiornare i limiti temporali di visita degli ordini già approvati. In particolare gli ordini precedenti all'ordine j nel percorso parziale potrebbero dover anticipare l'ultimo istante ammissibile in cui essere serviti mentre quelli successivi potrebbero dover posticipare il primo momento disponibile. Ovvero [3]:

$$\text{For } k = i - 1 \text{ to } 0,$$

$$l_k = \min(l_k, l_{k+1} - T_{k,k+1}).$$

$$\text{For } k = i \text{ to } n+1$$

$$e_k = \max (e_k, e_{k-1} + T_{k-1,k}) .$$

1.3 Home Delivery Problem with Time Slot Incentives

La pubblicazione di Campbell e Savelsbergh [2006] introduce un problema denominato “Home Delivery Problem With Time Slot Incentives”, il cui obiettivo è ridurre i costi di consegna sfruttando l'uso di incentivi per influenzare le scelte dei consumatori. Il lavoro è suddiviso in due parti principali. Nella prima si valuta la disponibilità di fasce orarie con relativo costo di inserimento per il nuovo ordine grazie ad una variante dell'algoritmo euristico degli stessi Campbell e Savelsbergh [2004]. In particolare si provano a costruire un certo numero di percorsi che rispettino i vincoli degli ordini già accettati e per ognuno di questi e per ogni fascia oraria si valuta l'inserimento del nuovo ordine e si memorizzano i costi di inserimento minori. Dopo che il cliente effettua la sua scelta viene aggiornato il percorso e quest'ultimo sarà il primo ad essere utilizzato per valutare i costi di inserimento dei nuovi ordini. La seconda parte introduce uno

schema di assegnamento degli incentivi. In particolare viene proposta una modellazione matematica per scegliere dinamicamente la quantità e il valore ottimale dell'incentivo da offrire al cliente. Per decidere a quali fasce orarie con probabilità positiva (insieme O) assegnare l'incentivo, questi vengono prima suddivisi in due gruppi:

- U , sottoinsieme di O di fasce orarie che potrebbero ricevere l'incentivo.
- V , sottoinsieme di O di fasce che non riceveranno l'incentivo.

Si vogliono determinare il valore dell'incentivo per la fascia oraria t , I^t , e la riduzione della probabilità per le fasce orarie in V , z , per massimizzare il profitto. La funzione obiettivo con relativi vincoli viene presentata nella Figura 2.

$$\begin{aligned} \max \quad & \sum_{t \in U} (r_i - C^t - I^t)(p_i^t + xI^t) + \sum_{t \in V} (r_i - C^t)(p_i^t - z) \\ \text{subject to:} \quad & z \leq p_i^t \quad \forall t \in V \\ & \sum_{t \in U} xI^t = z|V| \\ & 0 \leq I^t \leq B \quad \forall t \in U. \end{aligned}$$

Figura 2: funzione obiettivo per massimizzazione dei profitti. Campbell, Ann & Savelsbergh, Martin. (2006). Incentive Schemes for Attended Home Delivery Services. Transportation Science. 40. 327-341. 10.1287/trsc.1050.0136, p.332.

La prima sezione dell'obiettivo indica i profitti attesi dalle fasce orarie con incentivi, mentre la seconda i profitti attesi da quelle senza incentivi. I vincoli imposti determinano limiti al valore dell'incentivo (B) e al valore di influenza negativa del disincentivo sulla scelta del cliente. I componenti di U possono essere selezionati in base al costo di inserimento.

1.4 Foresight Policy

La ricerca di Campbell e Savelsbergh [2006] non considera informazioni previsionali e storiche nella valutazione dei costi delle fasce orarie. La pubblicazione di Yang, Strauss, Currie e Eglese [2013] studia in dettaglio il comportamento dei clienti e utilizza una politica previsionale basata sulla valutazione dell'inserimento del nuovo ordine in percorsi già conclusi. Il costo finale di inserimento è stimato come una combinazione lineare dei costi per i due insiemi di tragitti, C_H^s (percorsi attuali) e C_F^s (percorsi previsionali). Per i primi ordini l'aspetto previsionale è molto importante, mentre nelle fasi finali sono più rilevanti i percorsi aggiornati. Si utilizza quindi un peso w_j di C_H^s , definito come $w_j := j/J$, considerando $j-1$ il numero di ordini accettati fino a questo

momento e J come il numero massimo di prenotazioni atteso. Il costo di inserimento per l'ordine j e il timeslot s è il seguente [5]:

$$C_j^s = w_j * C_H^s + (1 - w_j) * C_F^s.$$

1.5 Problema dell'azienda

Planeat, l'azienda con cui si è collaborato, gestisce un portale che offre la possibilità di ordinare a domicilio ingredienti o piatti pronti con particolare attenzione al rispetto dell'ambiente. Attualmente l'azienda dispone di un deposito a Pavia e opera principalmente nelle zone di Pavia e Milano con l'impiego di due veicoli. La ditta si trova dunque ad affrontare le difficoltà del “problema dell'ultimo miglio” anche se attenuate dalla momentanea area servita (limitata alle due città). Durante i colloqui con l'azienda sono emersi i seguenti punti problematici: giorni e fasce orarie con poca o, al contrario, picchi di domanda e richieste difficilmente gestibili a causa della lontananza dagli ordini già accettati. Si è quindi concordato che un aspetto importante fosse capire come si potesse influenzare la scelta dei clienti con l'uso di incentivi e disincentivi.

2 Strumenti utilizzati

Questo capitolo è così organizzato: le sezioni dalla 2.1 alla 2.3 sono dedicate alla descrizione di JavaScript, la sezione 2.4 a JSON e la 2.5 alla spiegazione riguardo le scelte di questi linguaggi/formati. Nella sezione 2.6 e 2.7 vengono invece analizzate le API utilizzate.

2.1 Introduzione JavaScript

“JavaScript (spesso abbreviato in JS) è un linguaggio leggero, interpretato, funzionale e orientato agli oggetti, conosciuto per lo più come linguaggio di script per pagine web, ma utilizzato in molti ambienti non-browser [...]. Lo standard JavaScript è ECMAScript” [13]. JavaScript prevede sei tipi di dato primitivi, numeri, stringhe, booleani, simboli, *null* e *undefined*, e un tipo di dato complesso, gli oggetti. “Tutti gli altri elementi previsti dal linguaggio, come ad esempio gli array, le espressioni regolari, le funzioni, sono in realtà oggetti” [14]. JavaScript è un linguaggio debolmente tipizzato, ovvero il tipo delle variabili viene fissato al momento dell’esecuzione e può cambiare. Questa categoria di linguaggi necessita di un supporto run-time per i controlli in esecuzione con conseguente minor efficienza e sicurezza rispetto a linguaggi fortemente tipizzati, ma rende più agevole e flessibile la fase di programmazione.

2.2 Linguaggio di scripting e interpretato

“Un linguaggio di scripting, altro non è che un linguaggio di programmazione interpretato, destinato a compiti di automazione del sistema operativo (per esempio il batch) o delle applicazioni (per esempio le macro), o a essere usato all’interno delle pagine web (per esempio javascript o il più famoso jQuery)” [12]. I linguaggi interpretati offrono un’elevata portabilità, ma la necessità di una “traduzione” in esecuzione causa un aumento del lavoro del processore con conseguente ricaduta sulle prestazioni.

2.3 Linguaggio orientato agli oggetti

JavaScript è un linguaggio ad oggetti basato sul concetto di prototipo. “Introduce la nozione di oggetto prototipo (prototypical object), un oggetto usato come modello da cui prendere le proprietà iniziali per un nuovo oggetto. Ogni oggetto può specificare le sue proprietà, anche quando viene creato o in fase di esecuzione. Inoltre, ogni oggetto può essere associato ad un altro oggetto come prototipo, consentendo al secondo oggetto di condividere le proprietà del primo” [13]. Non vi è distinzione tra classi e istanze, ma vi sono solo oggetti. “Per creare oggetti con un particolare set di proprietà e valori si definisce una funzione costruttore. Ogni funzione JavaScript può essere usata

come costruttore” [13]. Inoltre è possibile gestire l’ereditarietà sempre tramite il concetto di prototipo. Infatti se ”si aggiunge una proprietà a un oggetto che è usato come prototipo per un gruppo di oggetti, anche gli oggetti del gruppo ereditano la nuova proprietà” [13]. A partire dalla versione ECMAScript 2015 “è prevista la possibilità di utilizzare un approccio sintattico alla definizione di oggetti analogo alle classi. [...] L’introduzione del costrutto *class* è solo una semplificazione sintattica, ma semanticamente JavaScript rimane un linguaggio senza classi” [14]. Oltre a questa semplificazione ECMAScript 2015 ha apportato ulteriori modifiche tra cui le costanti, i parametri con valori di default e variabili con visibilità di blocco.

2.4 JSON

“JSON (JavaScript Object Notation) è un semplice formato per lo scambio di dati. Per le persone è facile da leggere e scrivere, mentre per le macchine risulta facile da generare e analizzarne la sintassi. Si basa su un sottoinsieme del Linguaggio di Programmazione JavaScript, Standard ECMA-262 Terza Edizione - Dicembre 1999” [15]. JSON si è affiancato a XML come formato/serializzazione dei dati, risultando spesso un’ottima soluzione, data la sua semplicità e leggerezza. Nonostante derivi da JavaScript ne è completamente indipendente, ma la similarità della sintassi rende facilmente convertibili dati JSON in oggetti JavaScript.

2.5 Motivazione della scelta

JavaScript è stato scelto per la sua portabilità, caratteristica dei linguaggi interpretati, e per la facilità con cui permette di interagire con le API. In particolare lo scambio dei dati viene reso agevole dalla facilità di conversione dei dati JSON in oggetti JavaScript. Inoltre per le caratteristiche del progetto, per cui si rende necessaria l’interazione con l’utente per l’inserimento dei dati per la simulazione in un apposito modulo, lo sviluppo di una pagina web in HTML con JavaScript è stata ritenuta una buona soluzione per la semplicità con cui permette di gestire questo compito.

2.6 API

Le API utilizzate si suddividono in API per calcolare distanze geografiche e API per creare percorsi ottimizzati che visitino una e una sola volta un insieme di località. Le prime richiedono come input un insieme di origini e destinazioni e restituiscono le distanze (tempo e chilometri) basate sul più breve itinerario tra ogni origine e destinazione. Le seconde a partire da un insieme di coordinate, un certo numero di veicoli e vincoli, restituiscono il percorso ottimale con tempi e distanze di percorrenza. Vengono analizzate le seguenti API nella Tabella 1 e nella Tabella 2: RouteXL, Bing

Map Distance Matrix API, Bing Map Multi-Itinerary Optimization API, Google Distance Matrix API, Mapbox Optimization API e Mapbox Matrix API.

Tabella 1: descrizione Distance Matrix API. Vengono presentati i limiti giornalieri e totali di richieste e le modalità di calcolo per le seguenti API: RouteXL [7], Bing Map Distance Matrix API [9], Google Distance Matrix API [8] e Mapbox Matrix API [10].

DISTANCE MATRIX API	MAPS	BING MAPS	MAP BOX	ROUTE XL
RICHIESTE	40 000 richieste mensili (coppia origine-destinazione), ovvero circa 1300 al giorno.	125 000 richieste totali (per tutte le API), circa 340 al giorno. Una richiesta è intesa come 4 coppie della matrice.	100 000 richieste mensili, ovvero circa 3300 al giorno. Una richiesta è intesa come singolo elemento della matrice.	Massimo 100 richieste al giorno.
LIMITI	Massimo 25 origini o 25 destinazioni per richiesta.	Massimo 2500 coppie di origini-destinazioni, calcolata moltiplicando il numero di origini per il numero di destinazioni.	Massimo 25 coordinate per richiesta (10 se informazioni con traffico). Massimo 60 richieste per minuto (30 se informazioni con traffico).	Massimo 10 coordinate per richiesta.
CALCOLO	Distanza e tempo in base al percorso migliore. Può considerare il traffico.	Distanza e tempo in base al percorso migliore. Considera il traffico.	Distanza e tempo in base al percorso migliore. Può considerare il traffico.	Distanza considerando il percorso migliore (se possibile) o in linea d'aria.

Tabella 2: descrizione Optimization API. Vengono presentati i limiti giornalieri e totali di richieste e le modalità di calcolo per le seguenti API: RouteXL [7], Bing Map Multi-Itinerary Optimization API [9], e Mapbox Optimization API [10].

OPTIMIZATION API	MAPS	BING MAPS	MAP BOX	ROUTE XL
RICHIESTE	Non disponibile	125 000 richieste totali (per tutte le API), circa 340 al giorno. Il numero di transazioni è ottenuto come prodotto del numero di autisti e di fermate.	100 000 richieste mensili, ovvero circa 3300 al giorno. Una richiesta è intesa come singolo percorso.	Massimo 100 richieste al giorno.
LIMITI	Non disponibile	Massimo 3 agenti/veicoli e 20 richieste.	Massimo 12 coordinate per richiesta. Massimo 300 richieste per minuto.	Massimo 10 fermate.
CALCOLO	Non disponibile	Restituisce un percorso per uno o più veicoli per viaggiare tra più punti di consegna.	Restituisce un percorso ottimizzato per un veicolo. Non permette di specificare vincoli temporali per le fermate.	Percorso più veloce per uno o più veicoli.

2.7 API scelte per il progetto

Nello svolgimento del progetto sono state utilizzate RouteXL, Bing Map Distance Matrix API e Bing Map Multi-Itinerary Optimization API, in quanto Google non offre la possibilità di creare un percorso ottimizzato e Mapbox Optimization API non permette di specificare i vincoli temporali delle fasce orarie.

3 Progetto tesi

In questo capitolo si introduce il progetto (sezione 3.1) e si prosegue descrivendo gli elementi caratterizzanti nelle successive sezioni.

3.1 Introduzione

La proposta di tesi inizialmente era indirizzata verso il problema dell'ottimizzazione del trasporto. Una prima parte è stata quindi dedicata all'analisi dei software, delle API e delle pubblicazioni esistenti. Successivamente si è cominciato a discutere di come fosse possibile influenzare la scelta del cliente, con l'obiettivo primario di favorire richieste in una stessa zona e in una stessa fascia oraria e sfavorire richieste difficilmente gestibili, ad esempio in orari di punta. Per studiare più dettagliatamente il problema e confrontare possibili soluzioni si è deciso di sviluppare una piattaforma che simulasse l'arrivo di ordini in una giornata lavorativa. Le scelte del cliente vengono analizzate con e senza l'uso di incentivi e si valuta l'impatto finale in base al numero di ordini accettati e i tempi di percorrenza del tragitto finale.

3.2 Clienti

I clienti vengono posizionati nei pressi di Milano, generando delle coordinate con limiti imposti dalle posizioni dei comuni della città. Il comportamento del cliente viene modellato tramite probabilità, in particolare gli si può assegnare uno dei seguenti pattern di preferenza per le fasce orarie:

- per il cliente tutte le fasce orarie sono uguali
- per il cliente una fascia oraria ha il doppio della probabilità rispetto alle altre di essere scelta
- per il cliente una fascia oraria ha il triplo della probabilità rispetto alle altre di essere scelta.

Queste preferenze iniziali possono essere modificate solo tramite incentivi e disincentivi.

3.3 Fasce orarie

Le caratteristiche delle fasce orarie sono tra gli elementi più delicati da valutare. Infatti limitare la scelta del cliente verso un numero ridotto di fasce orarie potrebbe essere conveniente per l'azienda in quanto si riducono i tragitti dei veicoli, ma si ha un impatto negativo sull'esperienza del cliente. Viceversa un numero troppo elevato di fasce orarie potrebbe accontentare i clienti, ma essere ingestibile per l'azienda. Anche la durata temporale della fascia oraria può avere simili effetti. Nel progetto si lascia decidere entrambi questi parametri. Un altro fattore che si può inserire nella

simulazione è il “grado di occupazione” della fascia oraria, ovvero quanti ordini sono stati accettati rispetto ad un numero massimo di ordini che si ritiene di poter gestire in una singola fascia.

3.4 Incentivi

L’impiego di incentivi può avere il duplice effetto di far scegliere ai clienti, in particolare a quelli senza precise restrizioni in cui poter ricevere la consegna (ad esempio lavorative), di optare per una fascia oraria ritenuta la migliore per l’azienda e quello di evitare che il cliente decida di rinunciare ad effettuare l’ordine se non presente la fascia oraria desiderata. Gli incentivi possono assumere diverse forme: sconti sui costi di consegna, assegnamento di punti o anche segnalazione dell’impatto ambientale derivante dalla scelta di una fascia oraria. Nella simulazione non si considera un costo associato al loro utilizzo e, data l’importanza che l’azienda pone per l’aspetto ecologico, una soluzione potrebbe essere emulare il supermercato online britannico Ocado, il quale evidenzia con un simbolo (un camion verde) le fasce in cui è richiesta meno benzina per soddisfare l’ordine. Il progetto permette di analizzare i seguenti schemi di assegnamento di incentivi:

- assegnare incentivi alle fasce orarie più convenienti per l’azienda e disincentivi a quelle più onerose
- assegnare solo incentivi ridistribuendo la probabilità delle fasce orarie che non possono essere offerte
- assegnare solo incentivi e nel caso in cui tutte le fasce orarie siano servibili assegnare disincentivi.

Il valore percentuale di influenza dell’incentivo sulla scelta del cliente è un parametro da specificare all’avvio della simulazione. Va sottolineato che il numero di incentivi da destinare viene proporzionato al numero di fasce orarie disponibili per il cliente e azzerato se i costi di inserimento risultano equivalenti. Inoltre non vengono considerate fasce orarie con costo superiore ad un limite fissato.

3.5 Valutazione fattibilità fascia oraria e relativo costo: algoritmo euristico

Per valutare l’inserimento del nuovo ordine nel percorso parziale si segue una variante dell’algoritmo euristico di Campbell e Savelsbergh [2004] descritto nella sezione 1.2. Le distanze tra le coordinate vengono ottenute con una delle seguenti API : RouteXL o Bing Map Distance Matrix API. A differenza della soluzione sopra citata non si costruisce da zero un nuovo tragitto, ma si memorizza il percorso parziale degli ordini già accettati e si eseguono le fasi di valutazione della fattibilità, inserimento e aggiornamento solo per il nuovo ordine. Inoltre non si considerano i

profitti, ma i costi di inserimento calcolati come segue, con T ad indicare i tempi di percorrenza, j il nuovo ordine e $i-1$ e i gli ordini già accettati:

$$T_{i-1,j} + T_{j,i} - T_{i-1,i}.$$

Memorizzando per ogni ordine il primo e ultimo momento in cui può essere effettuata la consegna il costo computazionale dell'algoritmo si stima in $O(n^2)$, con n il numero di clienti. Infatti, per ogni cliente si deve valutare l'inserimento in ogni posizione possibile (ovvero tra ogni posizione dei clienti che hanno già completato l'ordine), operazione stimabile in $O(n)$. L'aggiornamento dei tempi ammissibili per le consegne precedenti e successive ha un ulteriore costo di $O(n)$. Quindi ognuna delle n iterazione principali ha un costo di $O(n)$ ottenendo complessivamente un costo di $O(n^2)$. Il calcolo del costo di inserimento e della fattibilità, date le distanze, si può invece considerare costante.

3.6 Valutazione fattibilità fascia oraria e relativo costo: API per percorsi ottimizzati

Lo schema della soluzione è come quello sopra descritto, ma per valutare le fasce orarie disponibili per il cliente e il loro costo ci si avvale dell'utilizzo di una tra le seguenti API, RouteXL e Multi-Itinerary Optimization API. Entrambe permettono di creare un percorso ottimizzato per un insieme di consegne, rispettando i vincoli temporali di ognuna di queste. La risposta consiste nel tragitto, con tempi e distanze da percorrere, o nell'indicazione della sua non realizzabilità. Per valutare la fattibilità di una nuova consegna si aggiunge al percorso parziale il nuovo ordine. A quest'ultimo si assegnano ciclicamente tutte le fasce orarie e si crea un nuovo percorso temporaneo. Per le fasce orarie in cui l'ordine può essere inserito correttamente si memorizza il costo di inserimento, sottraendo al nuovo tempo totale del tragitto quello precedente. Il costo computazionale ricade sull'esecuzione dell'API.

3.7 Previsione nella valutazione del costo di inserimento del nuovo ordine: Foresight Policy

Valutare il costo di inserimento del nuovo ordine solo in base agli ordini già accettati potrebbe non essere sufficiente, in particolar modo all'inizio della finestra temporale in cui possono essere effettuate le richieste. Diventa allora fondamentale una metodologia che consideri i "costi di opportunità". Questi derivano dall'accettare ordini che potrebbero causare la rinuncia ad alternative migliori, ad esempio per ricavi o facilità di gestione. Un primo approccio che può essere usato nelle simulazione è la "Foresight Policy" descritta nella pubblicazione di Yang, Strauss, Currie e Eglese

[2013] e presentata nella sezione 1.4. Per creare il percorso previsionale si utilizzano, per un diverso insieme di clienti, gli stessi parametri da usare nella simulazione del nuovo tragitto.

3.8 Previsione nella valutazione del costo di inserimento del nuovo ordine: peso fasce orarie

L'azienda ha fatto notare come alcuni giorni e fasce orarie siano particolarmente richiesti mentre altri siano poco ricercati. Questi ultimi in particolare comportano perdite per l'azienda, in quanto i mezzi a disposizione non vengono sfruttati. È stato quindi suggerito di utilizzare un peso per le fasce orarie che renda quelle statisticamente meno richieste quelle maggiormente da incentivare. La gestione di questo fattore nella simulazione avviene tramite un coefficiente da moltiplicare ai costi di inserimento dell'ordine nella fascia oraria. Ad ogni fascia oraria viene attribuito un peso (da 0 a 1), in particolare a quelle da incentivare viene assegnato un valore più elevato, e il coefficiente da utilizzare viene ottenuto sottraendo a 1 questa misura. Anche in questo caso l'informazione previsionale ha maggior significato per i primi ordini, per cui si utilizza, come nella "Foresight Policy" la variabile $w_j := j/J$, considerando $j-1$ il numero di ordini accettati fino a questo momento e J come il numero massimo di prenotazioni atteso. La formula finale per calcolare il costo di inserimento è la seguente, con le variabili *peso_timeslot*, *costo*, *n_orders_time* e *capacity* che rappresentano rispettivamente la priorità della fascia oraria, il costo di inserimento dell'ordine nella fascia oraria, il numero di ordini già accettati nella fascia oraria e il valore previsionale di ordini gestibili in una fascia oraria:

$$(1 - w) * (1 - \text{peso_timeslot}) + w * \{\text{costo} * [1 + (n_orders_time / \text{capacity})]\}$$

Si presentano nelle prossime pagine dei grafici ottenuti da un tool di disegno di funzioni online (Graphsketch [11]), utilizzati per analizzare la formula. I grafici presentano lungo l'ascissa il peso w e lungo l'ordinata il costo di inserimento dell'ordine nella fascia oraria (C).

Grafico

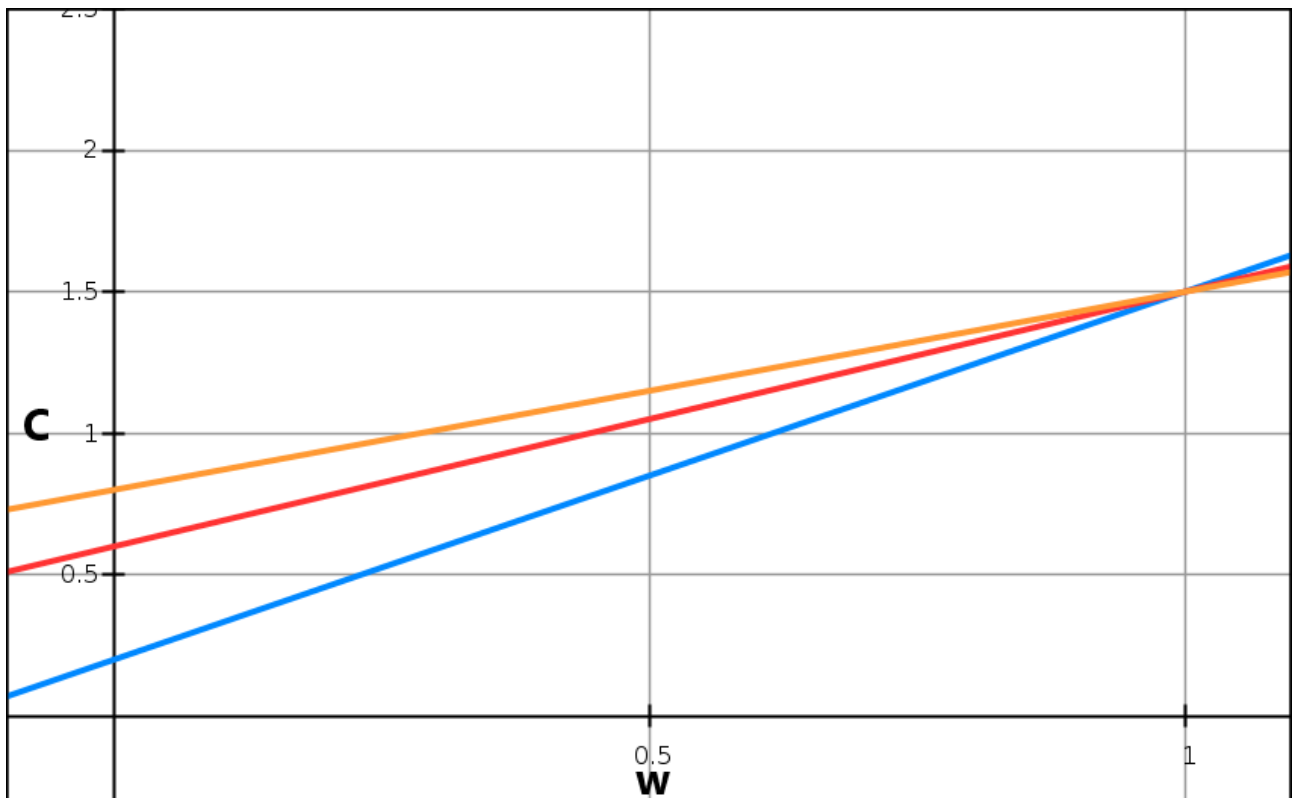


Figura 3: grafico di analisi del costo di inserimento del nuovo ordine applicando la politica con peso delle fasce orarie. I parametri sono tutti uguali ad eccezione dei pesi.

$Costo:=1$, $n_orders_time:=5$, $capacity:=10$

- $peso_timeslot:=0.8$
- $peso_timeslot:=0.4$
- $peso_timeslot:=0.2$

Nel grafico della Figura 3 si analizza l'impatto della priorità statica della fascia oraria. A parità degli altri parametri in ogni istante della finestra temporale di prenotazione sarebbero le fasce orarie scelte in base ai dati storici e statistici a venire incentivate.

Grafico

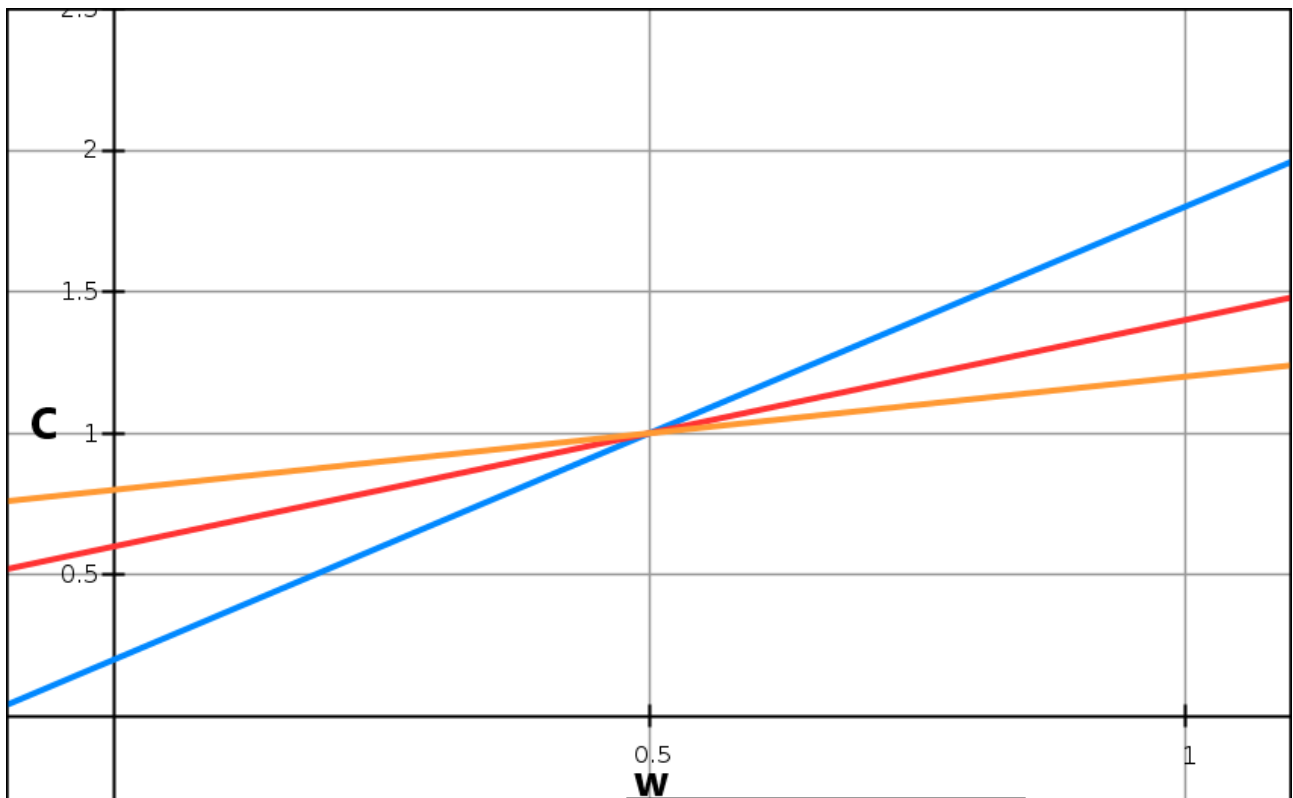


Figura 4: grafico di analisi del costo di inserimento del nuovo ordine applicando la politica con peso delle fasce orarie. Variano i pesi delle fasce orarie e il grado di occupazione.

Costo:=1, capacity:=10

- *peso_timeslot:=0.8, n_orders_time:=8*
- *peso_timeslot:=0.4, n_orders_time:=4*
- *peso_timeslot:=0.2, n_orders_time:=2*

Il grafico della Figura 4 analizza l'influenza sia della priorità della fascia oraria sia del suo "grado di occupazione". Si evince che all'aumentare del numero di ordini accettati l'informazione statica perde peso rispetto alle informazioni relative alla situazione attuale e nella parte finale della finestra temporale di prenotazione è la "saturazione" della fascia oraria a influenzare maggiormente il costo di inserimento dell'ordine.

Grafico

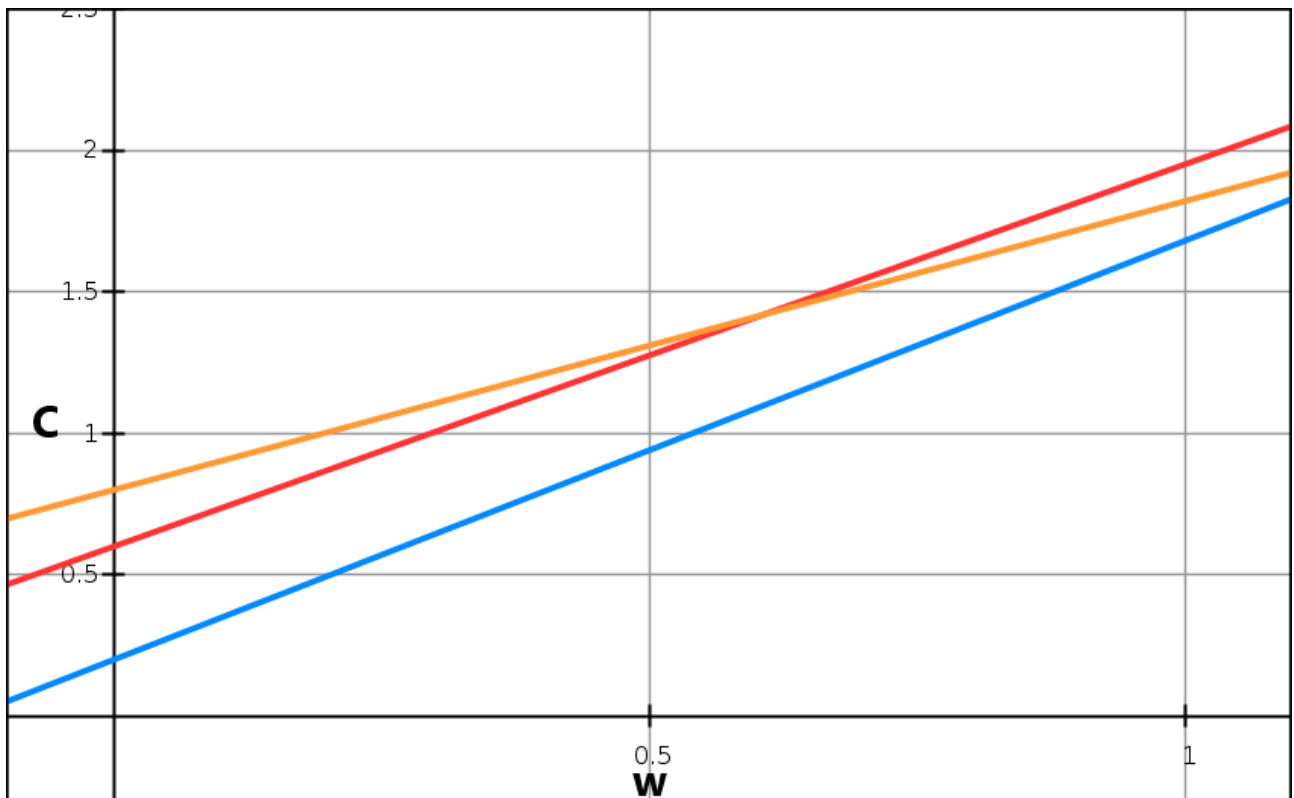


Figura 5: grafico di analisi del costo di inserimento del nuovo ordine applicando la politica con peso delle fasce orarie. Variano tutti i parametri ad eccezione del numero massimo di ordini che si ritiene di poter gestire in una singola fascia.

Capacity:=10

- *peso_timeslot:=0.8, n_orders_time:=4, cost:=1.2*
- *peso_timeslot:=0.4, n_orders_time:=5, cost:=1.3*
- *peso_timeslot:=0.2, n_orders_time:=3, cost:=1.4*

In questi ultimi grafici si analizza l'effetto del costo di inserimento. In particolare nel grafico di Figura 5 la soluzione con minor costo risulta la migliore in ogni fase della finestra temporale di prenotazione, in quanto possiede anche la maggior priorità.

Grafico

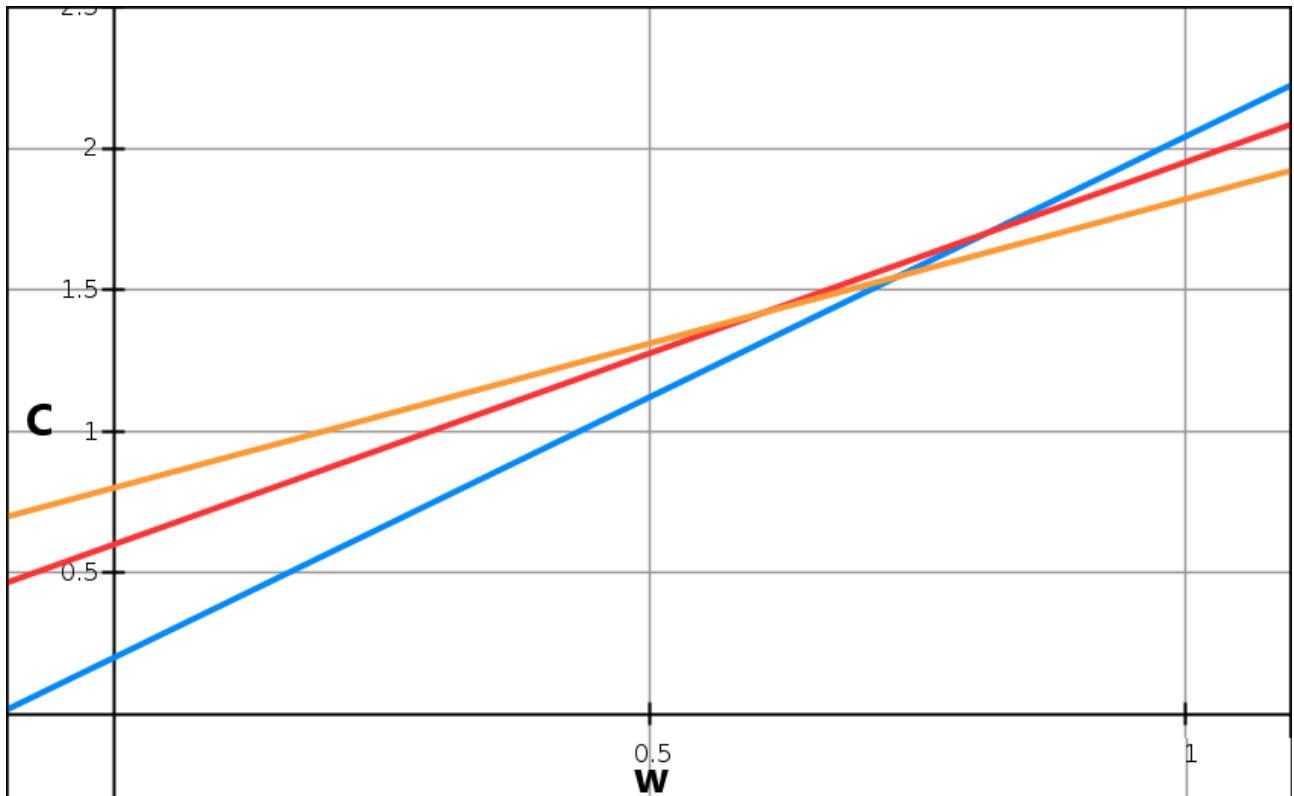


Figura 6: grafico di analisi del costo di inserimento del nuovo ordine applicando la politica con peso delle fasce orarie. Variano tutti i parametri ad eccezione del numero massimo di ordini che si ritiene di poter gestire in una singola fascia.

Capacity:=10

- $peso_timeslot:=0.8, n_orders_time:=7, cost:=1.2$
- $peso_timeslot:=0.4, n_orders_time:=5, cost:=1.3$
- $peso_timeslot:=0.2, n_orders_time:=3, cost:=1.4$

Nel grafico di Figura 6 si può vedere come un sovraccarico di ordini possa cambiare la situazione. Infatti per gli ultimi ordini la fascia migliore risulta quella più libera nonostante un costo iniziale più elevato.

4 Analisi simulazioni

Quest'ultimo capitolo inizia con l'analisi delle simulazioni (sezione 4.1) e si conclude con l'esame della politica di offerta delle fasce orarie ed evidenziando possibili sviluppi futuri (sezione 4.2 e 4.3).

4.1 Simulazioni

Nei test effettuati si valutano i risultati ottenuti utilizzando diversi schemi di incentivi rispetto ad analoghe condizioni senza l'applicazione degli stessi. Di seguito vengono indicati i tre schemi presenti nelle tabelle successive:

- NOINC (nessun incentivo): la scelta del cliente dipende unicamente dalle proprie preferenze iniziali verso le varie fasce orarie.
- INCDIS (incentivi e disincentivi): si applicano incentivi alle fasce orarie con minor costo di inserimento e disincentivi a quelle più onerose.
- ONLYINC (solo incentivi): si applicano solo incentivi alle fasce orarie più convenienti.

Quest'ultima strategia permette di analizzare anche l'impatto dei disincentivi. È stata introdotta in quanto si è ritenuto che se il cliente si trovasse in uno scenario in cui la fascia oraria preferita non fosse presente l'utilizzo di disincentivi potrebbe portarlo a non effettuare l'ordine. Al contrario gli incentivi potrebbero convincerlo a spostarsi in un'altra fascia. Di seguito sono invece presentati gli schemi previsionali:

- FP (foresight policy): si applica la "Foresight Policy" descritta nella pubblicazione di Yang, Strauss, Currie, Eglese [2013] e presentata nella sezione 1.4.
- WP (weight policy): si assegna alle fasce orarie una diversa priorità in base ad informazioni statistiche, come descritto nella sezione 3.8.
- NF (no forecast): non viene applicata nessuna previsione.

In tutti gli esperimenti, a meno che specificamente indicato, vengono generate le richieste di 20 clienti posizionati nella zona di Milano. Si utilizzano 2 veicoli e il tempo di servizio per completare la consegna una volta giunti dal cliente è di 10 minuti. I ricavi per ogni ordine sono di €3 e le fasce orarie hanno la durata di un'ora. Se l'inserimento del nuovo ordine aumenta i tempi di viaggio del percorso parziale oltre l'ora non vengono assegnati incentivi. Nelle simulazioni sono state utilizzate le API Bing Map, in quanto non si hanno restrizioni giornaliere, e si esegue l'algoritmo euristico presentato nella sezione 3.5 che comporta un minor numero di richieste. I parametri che possono variare riguardano il numero di fasce orarie e il numero massimo di incentivi assegnabili. Per ogni schema si effettuano simulazioni considerando tutte le fasce orarie identiche per il cliente (PROBPAT=1) e con preferenze diverse (PROBPAT=2). Si calcola la media tra i risultati di questi due casi e si considera la variazione percentuale sia rispetto ai dati ottenuti nelle stesse simulazioni sia per tutte le esecuzioni per uno schema previsionale, in entrambi i casi nello scenario senza incentivi.

Nella Tabella 3 si presentano i dati delle simulazioni con 4 fasce orarie con massimo 2 incentivi assegnabili.

Tabella 3: risultati delle simulazioni con 20 clienti, 4 fasce orarie e massimo 2 incentivi assegnabili. Ogni riga valuta una politica previsionale applicata ad ogni schema di assegnamento degli incentivi.

Forecast	PROBPAT	NOINC	INCDIS	ONLYINC
NF	1	32,53	35,8	37,45
	2	35,17	36,07	32,68
Media		33,85	35,93	35,06
% miglioramento		0	+13,9	-3,13
% miglioramento complessivo		0	+6,14	+3,57
FP	1	29,38	30,99	27,67
	2	30,72	32,48	31,13
Media		30,05	31,73	29,4
% miglioramento		0	+3,76	-0,4
% miglioramento complessivo		0	+5,59	-3,74
WP	1	30,73	35,8	31,04
	2	31,87	32,65	35,3
Media		31,30	34,22	33,17
% miglioramento		0	+12,76	+2,88
% miglioramento complessivo		0	+9,32	+5,97

Questi primi dati evidenziano che, con un ridotto numero di fasce orarie, l'utilizzo sia di incentivi sia di disincentivi (INCDIS) porta ai migliori risultati. Nella "Foresight Policy" (FP) vengono assegnati pochi incentivi probabilmente in quanto il numero limitato di fasce rende "saturo" il percorso previsionale rendendo difficile l'inserimento dei nuovi ordini. L'offerta solo di incentivi (ONLYINC) non risulta invece sufficiente. Con questa tecnica si mira infatti a non perdere l'emissione dell'ordine del cliente, ma le risposte sembrano indicare che in condizioni di maggior restrizione come questa sia più importante la qualità della scelta piuttosto che la quantità.

I risultati della Tabella 4 sono stati calcolati considerando 7 fasce orarie a disposizione del cliente con massimo 3 incentivabili.

Tabella 4: risultati delle simulazioni con 20 clienti, 7 fasce orarie e massimo 3 incentivi assegnabili. Ogni riga valuta una politica previsionale applicata ad ogni schema di assegnamento degli incentivi.

Forecast	PROBPAT	NOINC	INCDIS	ONLYINC
NF	1	45,57	52,42	50,22
	2	41,09	43,11	44,09
Media		43,33	47,76	47,15
% miglioramento		0	+13,98	+5,59
% miglioramento complessivo		0	+10	+8,81
FP	1	42,82	50,72	49,08
	2	47,46	41,9	53,17
Media		45,14	46,31	51,12
% miglioramento		0	+5,73	+10
% miglioramento complessivo		0	+2,59	+13,24
WP	1	46,84	52,1	49,71
	2	47,63	48,3	51,47
Media		47,23	50,2	50,59
% miglioramento		0	+7,17	+9,12
% miglioramento complessivo		0	+6,28	+7,11

I valori ottenuti indicano come in questo contesto l'uso di incentivi porti sempre a profitti maggiori. Infatti, per ogni ordine, aumentano le fasce orarie disponibili e le posizioni di inserimento ammissibili e la gestione degli incentivi viene favorita da questa maggior quantità di opzioni.

Si considerano ora i casi estremi con 30 e 10 clienti nella giornata lavorativa. Nella Tabella 5 si mostrano i dati delle simulazioni con 30 clienti e 4 fasce orarie.

Tabella 5: risultati delle simulazioni con 30 clienti, 4 fasce orarie e massimo 2 incentivi assegnabili. Ogni riga valuta una politica previsionale applicata ad ogni schema di assegnamento degli incentivi.

Forecast	PROBPAT	NOINC	INCDIS	ONLYINC
NF	1	33,72	33,68	29,05
	2	36,03	38,31	35,55
Media		34,87	35,99	32,3
% miglioramento		0	+6,82	-11,6
% miglioramento complessivo		0	+3,21	-7,95
FP	1	33,67	33,56	27,42
	2	35,94	30,65	36,76
Media		34,8	32,1	32,09
% miglioramento		0	-10,8	-6,23
% miglioramento complessivo		0	-8,4	-8,44
WP	1	32,09	37,67	33,68
	2	30,69	33,64	24,83
Media		31,39	35,65	29,25
% miglioramento		0	+4,96	+3,95
% miglioramento complessivo		0	+13,57	-7,31

I risultati confermano le conclusioni espresse nel caso precedente avente 20 clienti e 4 fasce orarie disponibili, con lo schema INCDIS ritenuto il più adatto. Va però sottolineato come i profitti si accostino alle analoghe esecuzioni con 10 clienti in meno. Con un numero limitato di fasce orarie si raggiunge quindi presto la saturazione e gli ordini della fase finale della finestra temporale di prenotazione difficilmente vengono accettati.

La Tabella 6 presenta lo scenario con 30 clienti e 7 fasce orarie.

Tabella 6: risultati delle simulazioni con 30 clienti, 7 fasce orarie e massimo 3 incentivi assegnabili. Ogni riga valuta una politica previsionale applicata ad ogni schema di assegnamento degli incentivi.

Forecast	PROBPAT	NOINC	INCDIS	ONLYINC
NF	1	52,61	63,41	57,43
	2	57,84	57,75	60,43
Media		55,22	60,58	58,93
% miglioramento		0	+5,76	+10,8
% miglioramento complessivo		0	+9,70	+6,71
FP	1	62,23	58,68	62,73
	2	57,63	60,9	66,53
Media		59,93	59,79	64,63
% miglioramento		0	-3,42	+11,39
% miglioramento complessivo		0	-0,2	+7,84
WP	1	58,82	63,56	58,55
	2	47,15	56,88	58
Media		52,98	60,22	58,27
% miglioramento		0	+18,96	+5,27
% miglioramento complessivo		0	+13,6	+9,98

Complessivamente anche in questa situazione si riconfermano le impressioni precedenti, sottolineando però che generalmente si ha un importante aumento dei profitti. Inoltre la politica basata sul “grado di occupazione” e diverse priorità delle fasce orarie (WP) ha registrato ottimi risultati.

Si presentano infine i risultati delle esecuzioni con 10 clienti con 4 (Tabella 7) e 7 (Tabella 8) fasce orarie disponibili. Non vi sono evidenze particolari rispetto ai casi precedenti se non l'aumento della percentuale di ordini accettati e gestibili anche con poche fasce offerte. Questo ha permesso anche alla politica basata solo sull'offerta di incentivi (ONLYINC) di avere un buon effetto sui profitti totali.

Tabella 7: risultati delle simulazioni con 10 clienti, 4 fasce orarie e massimo 2 incentivi assegnabili. Ogni riga valuta una politica previsionale applicata ad ogni schema di assegnamento degli incentivi.

Forecast	PROBPAT	NOINC	INCDIS	ONLYINC
NF	1	24,98	27,41	25,47
	2	23,26	26,5	25,68
Media		24,12	26,95	25,57
% miglioramento		0	+0,6	+19,15
% miglioramento complessivo		0	+11,73	+6,01
FP	1	22,70	20,69	24,98
	2	21,78	23,49	22,62
Media		22,24	22,09	23,8
% miglioramento		0	+1,46	+4,79
% miglioramento complessivo		0	-0,6	+7,01
WP	1	21,59	22,84	24,94
	2	21,69	23,69	20,71
Media		21,64	23,26	22,82
% miglioramento		0	+12,58	+0,8
% miglioramento complessivo		0	+7,48	+5,45

Tabella 8: risultati delle simulazioni con 10 clienti, 7 fasce orarie e massimo 3 incentivi assegnabili. Ogni riga valuta una politica previsionale applicata ad ogni schema di assegnamento degli incentivi.

Forecast	PROBPAT	NOINC	INCDIS	ONLYINC
NF	1	26,21	26,73	25,97
	2	24,29	25,6	26,68
Media		25,25	26,16	26,32
% miglioramento		0	+1,78	+6,12
% miglioramento complessivo		0	+3,60	+4,23
FP	1	24,15	25,73	25,71
	2	25,4	25,62	25,91
Media		24,77	25,59	25,81
% miglioramento		0	+1,74	+5,82
% miglioramento complessivo		0	+3,31	+4,19
WP	1	23,78	22,78	25,5
	2	26,21	20,77	25,93
Media		24,99	21,77	25,71
% miglioramento		0	-19,52	+7,25
% miglioramento complessivo		0	-14,79	+2,88

Si conclude questa sezione notando che le politiche previsionali adottate hanno portato nella maggior parte dei casi a risultati peggiori rispetto agli analoghi schemi senza il loro impiego. Le tecniche hanno però fondamenti e idee che nella realtà potrebbero cambiare questa situazione. Infatti si ritiene che sia l'assenza di dati storici effettivamente collegati all'esecuzione a causare queste risposte.

4.2 Intuizioni e analisi della politica di offerta delle fasce orarie

Gli esperimenti hanno mostrato come anche semplici schemi di assegnamento degli incentivi possono incidere sul percorso finale e sui profitti derivanti. In particolare si è potuto intuire che:

- In scenari con restrizioni importanti, ad esempio con poche fasce orarie offribili, è preferibile orientare la scelta del cliente verso gli orari migliori e dissuaderlo dai più difficilmente gestibili piuttosto che “insistere” affinché emetta l'ordine.
- Un adeguato numero di fasce orarie a disposizione permette di avere un maggior numero di opzioni per l'inserimento del cliente e quindi avere un impatto maggiore degli incentivi.
- Alcune simulazioni hanno portato a risultati peggiori rispetto alle analoghe esecuzioni senza incentivi. Questo evidenzia come le scelte non debbano essere fatte solo in base al momento

considerato e come sia fondamentale avere una accurata modellazione del cliente e adeguati dati previsionali.

- Con un elevato numero di ordini e poche fasce a disposizione diminuisce drasticamente la quantità di quelli accettati nella parte finale della finestra di prenotazione. Questo sottolinea come le scelte dovrebbero basarsi anche su previsioni della domanda per gestire al meglio le risorse e non dover rifiutare richieste consistenti.

Per quanto concerne la politica di offerta delle fasce orarie basata sulla valutazione della fattibilità (presentata nelle sezioni 3.5 e 3.6) ha portato ad ottimi risultati permettendo di accontentare (come mostrato in Figura 7), con l'eccezione dello scenario con 30 clienti e 4 fasce a disposizione, oltre il 60% dei clienti. Inoltre grazie a questa strategia, al termine delle prenotazioni, si ha la certezza che esista almeno un percorso che visiti tutti i clienti e che ne rispetti i vincoli. È fondamentale però individuare il corretto numero di fasce orarie in base alla domanda per poter accettare il maggior numero di ordini e non avere “sprechi” di risorse.

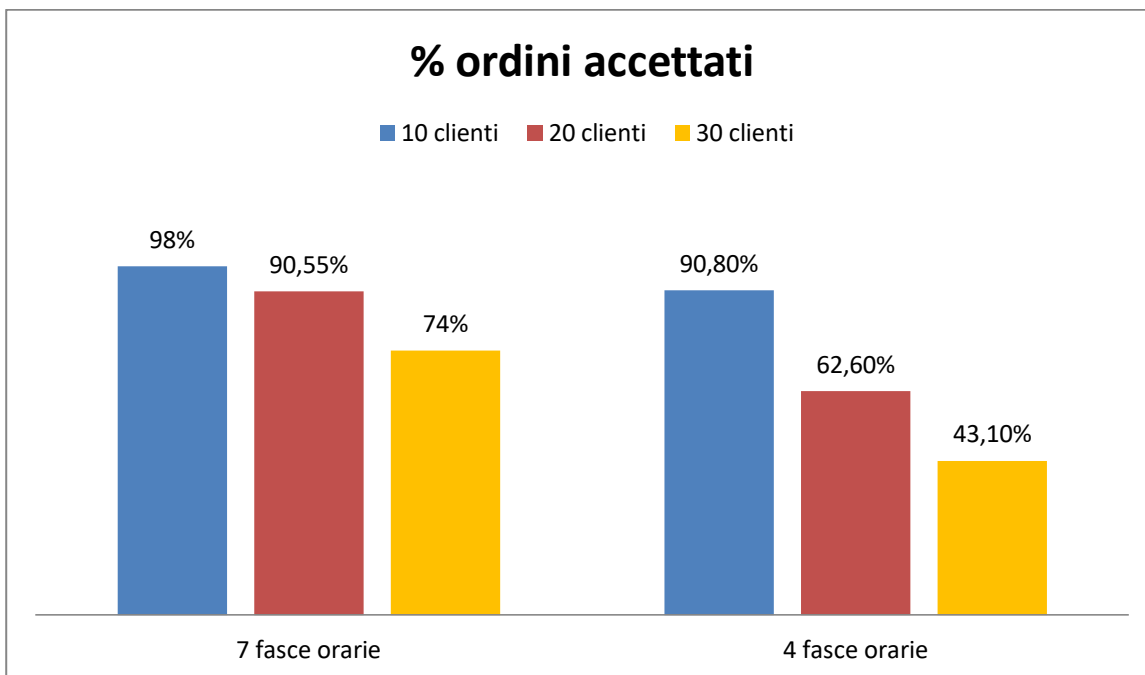


Figura 7: percentuale degli ordini accettati utilizzando la politica di offerta delle fasce orarie basata sulla valutazione della fattibilità dell'inserimento dell'ordine nel percorso parziale. A sinistra si presentano i risultati delle simulazioni con 7 fasce orarie e a destra quelli con 4 fasce orarie.

4.3 Conclusioni e futuri sviluppi

Il progetto presentato si propone di studiare come l'uso di incentivi e disincentivi possa influenzare il comportamento del consumatore per orientarlo verso le fasce orarie ritenute le più adatte alla creazione di un percorso finale ottimizzato per la giornata lavorativa. I risultati della simulazione sono però strettamente dipendenti dalla rappresentazione dei clienti data e dai parametri scelti. Un primo punto di approfondimento necessario riguarda quindi la modellazione delle scelte del cliente. Nel lavoro svolto si segue la linea di Campbell e Savelsbergh [2006] per cui si suppone di

conoscere le preferenze del cliente per gli orari di consegna. Una soluzione alternativa viene data da Yang, Strauss, Currie e Eglese [2013] con una modellazione denominata MNL (multinomial logit) basata su dati storici di prenotazione. Anche le tecniche usate nella gestione degli incentivi devono essere considerate solo come basi per evoluzioni più sofisticate. In particolare non si considerano costi legati all'utilizzo dell'incentivo, mentre uno dei principali metodi per applicarlo consiste nel destinare sconti sui costi di consegna. Una possibile strategia è seguire la modellazione illustrata da Campbell e Savelsbergh [2006] e descritta nella sezione 1.3. Inoltre le pubblicazioni consultate si basano sulla gestione di più tragitti da utilizzare per la valutazione del nuovo ordine. Questa politica innalza le possibilità di individuare fasce di consegna disponibili per il cliente, ma accresce i tempi di esecuzione che, in un utilizzo reale, devono essere di pochi secondi. Per seguire questa struttura si renderebbe necessaria una costruzione parallela dei percorsi. Come ultimo aspetto bisogna considerare che il lavoro è basato sulla situazione attuale dell'azienda che opera in una zona abbastanza limitata. Un'ulteriore analisi dovrebbe studiare possibili evoluzioni per capire quali combinazioni di vincoli utilizzare e valutare se integrare in un'unica soluzione la gestione di più zone e depositi o se suddividere il problema e trattarli indipendentemente.

BIBLIOGRAFIA

- [1] Metodi Esatti di Risoluzione per il Problema di Vehicle Routing con Finestre Temporali. Marina Malatesta, 2003-2004.
- [2] Tecniche di programmazione ed ottimizzazione dei servizi di trasporto merci e logistica, Daniele Vigo.
- [3] Campbell, Ann & Savelsbergh, Martin. (2004). Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. Transportation Science. 38. 369-378. 10.1287/trsc.1030.0046.
- [4] Ann Melissa Campbell and Martin Savelsbergh. Incentive schemes for attended home delivery services:. Transportation Science, 40:327-341, 2006.
- [5] Yang, Xinan & Strauss, Arne & Currie, Christine & Eglese, Richard. (2014). Choice-Based Demand Management and Vehicle Routing in E-Fulfillment. Transportation Science. 140807110218007. 10.1287/trsc.2014.0549.

SITOGRAFIA

- [6] Wikipedia: <https://it.wikipedia.org/>
- [7] RouteXL: <https://www.routexl.com/>
- [8] Google Routes: <https://developers.google.com/maps/>
- [9] Bing Maps: <https://www.microsoft.com/en-us/maps>
- [10] Mapbox: <https://www.mapbox.com/>
- [11] Graphsketch: <https://graphsketch.com/>
- [12] Insidevcode: <https://www.insidevcode.eu/2015/01/14/linguaggi-di-scripting-ecco-qui-alcuni/>, Flavius Florin Harabor.
- [13] Developer.mozilla: https://developer.mozilla.org/it/docs/Web/JavaScript/Guida/Dettagli_Object_Model
- [14] Html.it: <https://www.html.it/pag/55411/prototipi-e-classi/> , Andrea Chiarelli, 12 gennaio 2016.
- [15] JSON: <https://www.json.org/json-it.html>