



Università degli studi di Padova

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

Corso di Laurea Magistrale in Ingegneria Gestionale

**TECNICHE EURISTICHE DI
CLUSTERIZZAZIONE E PERCORSO OTTIMO
PER RISOLVERE UN PROBLEMA DI
RACCOLTA E CONSEGNA DI MATERIALE**

RELATORE: CH.MO PROF. GIORGIO ROMANIN JACUR

LAUREANDO: ADRIANO CORRENTE

ANNO ACCADEMICO: 2013 / 2014

*A mio padre,
maestro di vita*

INDICE

INDICE	3
SOMMARIO	7
INTRODUZIONE	9
CAPITOLO 1	11
LA RAPPRESENTAZIONE DEL PROBLEMA ATTRAVERSO I GRAFI	11
1.1.DEFINIZIONE DI GRAFO	11
1.2.PROPRIETÀ DEI GRAFI	12
1.2.1. <i>Grafi non orientati</i>	12
1.2.2. <i>Grafi orientati</i>	13
1.3.RAPPRESENTAZIONE DEI GRAFI	13
1.3.1. <i>Lista di adiacenza</i>	14
1.3.2. <i>Matrice di adiacenza</i>	15
1.3.3. <i>Matrice di incidenza</i>	16
CAPITOLO 2	17
VEHICLE ROUTING PROBLEM (VRP)	17
2.1.DEFINIZIONE DI VRP E CONCETTI GENERALI	17
2.1.1. <i>Depositi</i>	18
2.1.2. <i>Veicoli</i>	18
2.1.3. <i>Clienti</i>	19
2.1.4. <i>Obiettivi e Vincoli</i>	19
2.2.CLASSIFICAZIONE E VARIANTI PRINCIPALI DEL VRP	20
2.2.1. <i>Capacitated Vehicle Routing Problem (CVRP)</i>	21
2.2.2. <i>Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)</i>	21
2.2.3. <i>Distance-Constrained Vehicle Routing Problem (DCVRP)</i>	21
2.2.4. <i>Vehicle Routing Problem with Backhauls (VRPB)</i>	22
2.2.5. <i>Vehicle Routing Problem with Pickup and Delivery (VRPPD)</i>	22
2.3.CAPACITATED VEHICHI ROUTING PROBLEM WITH TIME WINDOWS (CVRPTW).....	23
2.3.1. <i>Aspetti generali del CVRPTW e vincoli</i>	23
2.3.2. <i>Definizione del problema e obiettivo</i>	24
CAPITOLO 3	27
TECNICHE EURISTICHE PER LA RISOLUZIONE DI UN VRP	27
3.1.CLASSIFICAZIONE TECNICHE EURISTICHE	27

3.1.1.	<i>Euristici costruttivi</i>	27
3.1.2.	<i>Euristici a due fasi</i>	28
3.1.3.	<i>Euristici migliorativi</i>	28
3.2.	TIPOLOGIE DI TECNICHE EURISTICHE	28
3.2.1.	<i>Savings Heuristics</i>	28
3.2.2.	<i>A Time-Oriented, Nearest-Neighbor Heuristic</i>	29
3.2.3.	<i>Insertion Heuristics</i>	30
3.2.3.1.	<i>Insertion Heuristic I1</i>	30
3.2.3.2.	<i>Insertion Heuristic I2</i>	31
3.2.3.3.	<i>Insertion Heuristic I3</i>	31
3.2.4.	<i>A Time-Oriented Sweep Heuristic</i>	32
CAPITOLO 4		33
CVRPTW: RACCOLTA, TRASPORTO E CONSEGNA DELLE PELLI		33
4.1.	DESCRIZIONE DEL PROBLEMA	33
4.1.1.	<i>Soggetti che contraddistinguono il problema</i>	34
4.1.1.1.	<i>Deposito</i>	35
4.1.1.2.	<i>Macelli</i>	35
4.1.1.3.	<i>Conceria</i>	35
4.1.1.4.	<i>Rete stradale</i>	35
4.2.	DATI, INDICI E NOTAZIONI SPECIFICHE DEL PROBLEMA	36
4.3.	FUNZIONE OBIETTIVO E VINCOLI DEL PROBLEMA	37
4.3.1.	<i>Funzione obiettivo</i>	37
4.3.2.	<i>Vincoli del problema</i>	37
4.3.3.	<i>Variabili del problema</i>	38
CAPITOLO 5		41
L'ALGORITMO EURISTICO CLUSTER PER LA SOLUZIONE DEL CVRPTW		41
5.1.	CVRPTW CON DEPOSITI MULTIPLI E VEICOLI ETEROGENEI	41
5.1.1.	<i>Definizione del problema</i>	41
5.1.2.	<i>Variabili decisionali e formulazione matematica del problema</i>	43
5.1.3.	<i>Vincoli del problema</i>	44
5.1.4.	<i>Regole di eliminazione esatta</i>	47
5.2.	ALGORITMO EURISTICO CLUSTER.....	49
5.2.1.	<i>Procedura definizione cluster</i>	49
5.2.1.	<i>Spiegazione dettagliata dell'algoritmo</i>	51
CAPITOLO 6		53
L'ALGORITMO EURISTICO CHEAPEST INSERTION PER LA SOLUZIONE DEL CVRPTW		53

6.1.INIZIALIZZAZIONE	53
6.2.ASSEGNAZIONE	55
6.3.VERIFICA DEI VINCOLI	59
6.4.SOLUZIONE DEL CVRPTW E CALCOLO DEL COSTO FINALE TOTALE	60
CAPITOLO 7.....	63
IMPLEMENTAZIONE IN EXCEL – VBA DELL’ALGORITMO CLUSTER CON CHEAPEST INSERTION	63
7.1.INSERIMENTO DATI DEL PROBLEMA RELATIVO A 13 MACELLI E 3 CAMION	64
7.1.1.Dati riguardanti i Macelli e la Conceria	65
7.1.2.Dati riguardanti i Camion	66
7.1.3.Dati riguardanti i Tempi di carico e scarico	68
7.1.4.Dati riguardanti i Costi di carico e scarico.....	68
7.1.5.Dati riguardanti i Tempi di viaggio per ogni camion.....	69
7.1.6.Dati riguardanti i Costi di viaggio per ogni camion	70
7.2.APPLICAZIONE ALGORITMO CLUSTER	72
7.3.APPLICAZIONE ALGORITMO CHEAPEST INSERTION	75
7.3.1.Predisposizione variabili	75
7.3.2.Fase di inizializzazione applicata ad un cluster specifico	76
7.3.3.Fase di assegnazione applicata ad un cluster specifico.....	78
7.3.4.Verifica dei vincoli e visualizzazione dati di elaborazione.....	81
7.3.5.Calcolo costi finali del tragitto ottenuto in un cluster specifico	83
7.4.SOLUZIONE FINALE DEL PROBLEMA	85
7.5.PROVA CON 20 MACELLI E 5 CAMION	85
7.5.1.Inserimento dati.....	85
7.5.1.1.Dati riguardanti i Macelli e la Conceria.....	85
7.5.1.2.Dati riguardanti i Camion	86
7.5.1.3.Dati riguardanti i Tempi di carico e scarico.....	87
7.5.1.4.Dati riguardanti i Costi di carico e scarico	87
7.5.1.5.Dati riguardanti i Tempi di viaggio per ogni camion	88
7.5.1.6.Dati riguardanti i Costi di viaggio per ogni camion	90
7.5.2.Applicazione algoritmo Cluster.....	92
7.5.3.Applicazione algoritmo Cheapest Insertion.....	93
7.5.4.Soluzione finale del problema	95
CONCLUSIONI	97
BIBLIOGRAFIA	99
APPENDICE	101
A.1.CODICE VBA USATO NEL PROGETTO	101
A.1.1.Dichiarazione delle variabili usate nell'applicativo	101

<i>A. 1.2. Interfaccia utente</i>	103
<i>A. 1.3. Predisposizione foglio di lavoro per input dati</i>	103
<i>A. 1.4. Gestione tabelle per l'inserimento dati</i>	104
<i>A. 1.5. Algoritmo Cluster</i>	121
<i>A. 1.6. Algoritmo Cheapest Insertion</i>	131
A. 1.6.1. Predisposizione variabili	131
A. 1.6.2. Inizializzazione.....	133
A. 1.6.3. Assegnazione	136
A. 1.6.4. Verifica dei vincoli e visualizzazione dati di elaborazione	144
A. 1.6.5. Costi finali del tragitto.....	150
A.2. TABELLE DEL PROBLEMA CON 13 MACELLI E 3 CAMION	157
<i>A.2.1. Dati di Input</i>	<i>157</i>
A.2.1.1. Tempi di viaggio per ogni camion	157
A.2.1.2. Costi di viaggio per ogni camion	158
<i>A.2.2. Tabelle di Elaborazione dati</i>	<i>160</i>
A.2.2.1. Dati elaborazione Tragitto 1.....	160
A.2.2.2. Dati elaborazione Tragitto 2.....	164
A.3. TABELLE DEL PROBLEMA CON 20 MACELLI E 5 CAMION	167
<i>A.3.1. Dati di Input</i>	<i>167</i>
A.3.1.1. Tempi di viaggio per ogni camion	167
A.3.1.2. Costi di viaggio per ogni camion	177
<i>A.3.2. Tabelle di Elaborazione dati</i>	<i>187</i>
A.3.2.1. Dati elaborazione Tragitto 1.....	187
A.3.2.2. Dati elaborazione Tragitto 2.....	190
A.3.2.3. Dati elaborazione Tragitto 3.....	194
RINGRAZIAMENTI	197

SOMMARIO

L'obiettivo di questa tesi è quello di approfondire il *Vehicle Routing Problem (VRP)* analizzando la progettazione di percorsi ammissibili per gruppi di veicoli che devono servire un certo numero di clienti. L'attenzione viene posta su importanti concetti quali le capacità limitate dei mezzi a disposizione e le finestre temporali; tali aspetti limitano fortemente il VRP e ne rendono complessa la risoluzione.

In questo elaborato viene presentato il *Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)* applicato ad uno specifico caso studio di consegna di pelli fresche ad una conceria dopo averle raccolte presso diversi macelli usando mezzi a capacità limitata. Questo problema di raccolta e consegna di materiale viene risolto con un metodo euristico a due fasi: nella prima viene applicata una tecnica euristica di clusterizzazione, al fine di accorpate i macelli in gruppi fattibili e convenienti, mentre nella seconda una tecnica euristica di inserzione (*Cheapest Insertion*), per trovare il sequenziamento ottimale dei macelli all'interno di ogni cluster.

La finalità è quella di costruire e assegnare i diversi percorsi ai veicoli a disposizione, i quali devono soddisfare per intero la domanda dei clienti e al tempo stesso generare il minor costo complessivo. La soluzione che porta a tale obiettivo è buona e ammissibile, in quanto ottenuta mediante l'impiego di tecniche euristiche e non esatte, garantendo al tempo stesso il rispetto dei vincoli. Per implementare la procedura, che permette in brevi tempi di elaborazione di ottenere tale soluzione, è stato usato il linguaggio di programmazione *Visual Basic for Applications (VBA)* in ambiente *Microsoft Office* (pacchetto applicativo *Excel*).

INTRODUZIONE

Nella logistica e nell'ambito dei trasporti, un aspetto fondamentale da considerare è la gestione efficiente della movimentazione dei veicoli, adibiti al servizio di raccolta/consegna di materiali ad un certo numero di clienti caratterizzati da una domanda nota. Dopo opportuna analisi, le decisioni in merito al numero e alla tipologia di veicoli da impiegare fra quelli a disposizione, al sequenziamento dei clienti seguito da ogni mezzo per soddisfare le richieste degli stessi, diventano concetti determinanti che, se ben applicati, mirano alla riduzione massima dei costi di trasporto. In questo ambito quindi, gestire le informazioni in tempi brevi, ottimizzare le scelte, ridurre le percentuali di mezzi che circolano parzialmente vuoti ed il numero dei veicoli circolanti, è di fondamentale importanza. Il *Capacitated Vehicle Routing Problem with Time Windows* risulta di grande interesse perché adattabile a diverse situazioni reali, in quanto tratta l'organizzazione e l'assegnazione di tragitti che alcuni mezzi a capacità limitata, partendo da un determinato deposito che può variare da mezzo a mezzo, devono percorrere al fine di servire ciascun cliente nel rispetto dei vincoli temporali prestabiliti.

Nel caso studio di questo elaborato viene affrontato il problema della raccolta delle pelli fresche che devono essere caricate nei macelli e trasportate alla conceria per poter essere ulteriormente lavorate. I mezzi, a capacità limitate, sono situati presso uno o più depositi, ciascuno caratterizzato da una finestra temporale che sancisce l'arco di tempo all'interno del quale il veicolo può iniziare il tragitto assegnato. Analogo concetto vale per i macelli e la conceria, in quanto ognuno di questi è contraddistinto da specifici orari di apertura e di chiusura, all'interno dei quali deve avvenire il servizio richiesto. Un ulteriore vincolo temporale è determinato dalla deperibilità del materiale trasportato: le pelli raccolte presso i macelli devono raggiungere la conceria entro tempi stabiliti al fine di evitare il loro deperimento e conseguente inutilizzo. Questo problema, caratterizzato da determinati dati di input e da specifici vincoli da rispettare, viene risolto con un metodo euristico a due fasi: nella prima viene applicata una tecnica euristica di clusterizzazione, al fine di accorpare i macelli in gruppi fattibili e convenienti, mentre nella seconda una tecnica euristica di inserzione (*Cheapest Insertion*), per trovare il sequenziamento ottimale dei macelli all'interno di ogni cluster.

Le diverse tecniche esatte, riportate in letteratura, richiedono elevati tempi di elaborazione e limitate dimensioni del problema trattato, a differenza delle tecniche euristiche. Queste, infatti, hanno generalmente limitati tempi di risposta che permettono

il conseguimento di buone soluzioni ammissibili, vicine all'ottimo, in tempi di calcolo accettabili e non presentano problemi dimensionali. Inoltre, gli algoritmi esatti, per essere sviluppati, necessitano dell'utilizzo di software specifici e dedicati, non di uso comune ma bensì legati a costose licenze per l'impiego, a differenza degli algoritmi basati su tecniche euristiche che possono essere implementati mediante software poco costosi, a larga diffusione e molto conosciuti. Rientra in quest'ultimo caso lo sviluppo del metodo euristico a due fasi che porta, in pochi istanti di elaborazione, alla soluzione del *Capacitated Vehicle Routing Problem with Time Windows* di questo elaborato, in quanto eseguito in *Visual Basic for Applications* nel pacchetto applicativo *Microsoft Excel*.

Questa tesi contiene al suo interno la spiegazione dettagliata del problema affrontato e ne permette la sua completa comprensione, partendo dal primo capitolo in cui sono esposti i grafi e le loro proprietà principali, elementi utili a rappresentare la soluzione di un *Vehicle Routing Problem (VRP)*, i cui concetti teorici generali sono riportati nel secondo capitolo, assieme alle sue principali varianti e classificazioni. L'attenzione viene focalizzata sul *Capacitated Vehicle Routing Problem with Time Windows*, oggetto di studio in questo elaborato. Nel terzo capitolo sono indicate le principali tecniche euristiche che permettono di ottenere una soluzione buona e ammissibile di un *VRP*, mentre nel quarto sono descritti l'ambiente del problema, i soggetti coinvolti, i dati di input ed i vincoli che lo contraddistinguono. Nel quinto e nel sesto capitolo sono riportate rispettivamente le spiegazioni puntuali dell'algoritmo euristico *Cluster* e dell'algoritmo euristico di *Cheapest Insertion* per ottenere la soluzione del *CVRPTW* con depositi multipli e veicoli eterogenei. Queste due procedure rappresentano la parte più importante di questa tesi e assieme costituiscono il metodo euristico a due fasi mediante il quale prima sono raggruppati i clienti in cluster e poi sono disposti in sequenza all'interno di ogni gruppo precedentemente individuato. Nel settimo capitolo è descritta l'implementazione in *Visual Basic for Applications* degli algoritmi di clusterizzazione e di *Cheapest Insertion*, partendo dalla gestione dell'inserimento dei dati del problema per arrivare alla sua soluzione finale. L'intera procedura è applicata inizialmente ad un contesto pratico composto da uno o più depositi, da cui possono partire 3 veicoli per soddisfare i fabbisogni e le richieste di 13 macelli. Successivamente l'analisi viene estesa ad un problema di routing più complesso, i cui soggetti coinvolti sono costituiti da 5 mezzi e 20 macelli. Il listato del codice *VBA* utilizzato in questo progetto e le tabelle che riportano i dati intermedi dell'elaborazione sono riportati in appendice.

CAPITOLO 1

La rappresentazione del problema attraverso i grafi

I grafi sono strutture matematiche discrete che rivestono interesse sia per la matematica che per un'ampia gamma di campi applicativi attinenti a molteplici aree quali, solo per citarne alcune, topologia, informatica (schematizzazione di programmi, circuiti, reti di computer, mappe di siti), chimica e organizzazione aziendale.

I grafi sono alla base di modelli di sistemi e processi studiati nell'ingegneria, nella ricerca operativa e nella geografia applicata all'urbanistica (reti stradali, cartografia e trasporti). Essi si prestano quindi a rappresentare problemi apparentemente molto diversi tra di loro con un linguaggio semplice e unificato.

Nei paragrafi seguenti vengono presentate e spiegate le principali definizioni e proprietà relative agli elementi grafici utili per rappresentare il *Vehicle Routing Problem* proposto in questo elaborato (Fischetti, 1999).

1.1. Definizione di grafo

Un grafo è un insieme di elementi detti *nodi* o *vertici*, (rappresentati generalmente da cerchi), che possono essere collegati fra loro da linee chiamate *lati* o *archi* o *spigoli*. Formalmente, un grafo G è una coppia ordinata $G = (V, E)$ in cui V identifica un insieme di nodi ed E rappresenta un insieme di coppie di elementi di V costituenti collegamenti tra vertici.

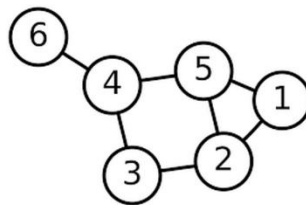


Figura 1.1 Esempio di grafo

Dato il lato $e = (u, v)$ i nodi u e v sono detti *estremi* di e , e si dice che il lato e , identificato con la coppia formata dai suoi estremi (u, v) , incide su u e su v .

1.2. Proprietà dei grafi

In base alla tipologia di grafo in esame, a ciascun arco può essere associato un *peso* (o un *costo*). Nella trattazione seguente vengono esplicate le proprietà dei grafi tenendo in considerazione solamente *grafi semplici*, cioè senza archi paralleli (coppie di archi aventi gli stessi estremi) e senza loop (archi di tipo (u, u)). Vengono subito distinti due tipi fondamentali di grafi: *orientati* e *non orientati*.

1.2.1. Grafi non orientati

Un grafo non orientato $G = (V, E)$ è definito da un insieme finito $V(G) = \{v_1, v_2, \dots, v_n\}$ di elementi detti vertici e da un insieme $E(G) = \{e_1, e_2, \dots, e_m\}$ di coppie non ordinate di nodi detti lati (o spigoli) (vedi figura 1.2). Dato lo spigolo $e = (u, v)$, la connessione che intercorre tra il nodo u ed il nodo v ha la stessa valenza della connessione che intercorre tra v ed u ; si ha quindi che $e = (u, v) = (v, u)$.

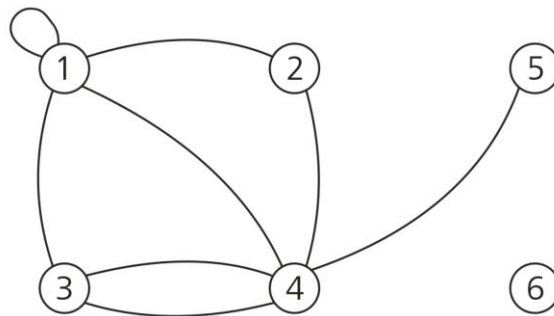


Figura 1.2 Esempio di grafo non orientato

Due vertici si dicono *adiacenti* se esiste il lato che li collega, mentre due spigoli si dicono *adiacenti* se hanno un vertice in comune.

Il *grado* del vertice v , di solito indicato con $d_g(v)$, è dato dal numero di lati incidenti nel vertice v stesso. Nel caso in cui $d_g(v) = 0$ si riscontra che v è un nodo *isolato* (vedi nodo 6 in figura 1.2) mentre nel caso di $d_g(v) = 1$ si può definire v come vertice *pendente* (vedi nodo 5 in figura 1.2).

Per ogni grafo si può definire un *percorso*, ossia una sequenza di vertici v_0, v_1, \dots, v_n unita ad una sequenza di lati che li collegano $(v_0, v_1), (v_1, v_2), \dots, (v_{n-1}, v_n)$. I vertici v_0 e v_n sono chiamati *estremi* del percorso.

Un percorso che al proprio interno non presenta lati ripetuti viene definito *cammino*; se quest'ultimo è *chiuso* (cioè presenta i nodi estremi coincidenti) allora prende il nome di *ciclo* o *circuito*.

1.2.2. Grafi orientati

Un grafo orientato $G = (V, A)$ è definito da un insieme finito $V(G) = \{v_1, v_2, \dots, v_n\}$ di elementi detti vertici e da un insieme $A(G) = \{a_1, a_2, \dots, a_m\}$ di coppie ordinate di nodi dette archi (vedi figura 1.3). Dato l'arco $a = (u, v)$, la connessione che intercorre tra il nodo u ed il nodo v non ha la stessa valenza della connessione che intercorre tra v ed u ; si ha quindi che l'arco $a_1 = (u, v)$ è distinto dall'arco $a_2 = (v, u)$.

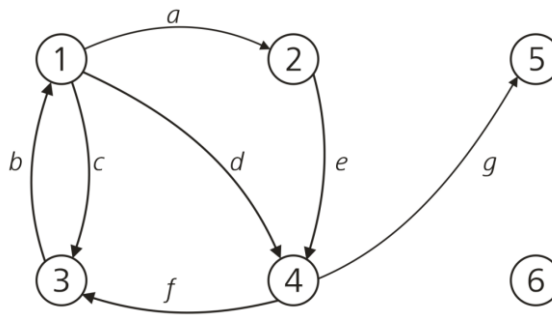


Figura 1.3 Esempio di grafo orientato

L'arco $a = (u, v)$ di un grafo orientato è caratterizzato da una *direzione*, rappresentata graficamente da una freccia uscente dal primo nodo della coppia (u) detto *coda*, ed entrante nel secondo nodo della coppia (v) detto *testa*.

Il *grado* del vertice v , di solito indicato con $d_g(v)$, è dato dal *grado entrante* (numero di archi entranti nel nodo) sommato al *grado uscente* (numero di archi uscenti dal nodo). Nella figura 1.3, per esempio, il vertice 3 ha grado uguale a 3, dato dalla somma del grado uscente (1) e del grado entrante (2). Restano invariati i concetti di nodo isolato e pendente come espresso nel precedente paragrafo.

Nel caso di grafo orientato si parla invece di *cammino orientato*, dato dalla sequenza di archi consecutivi del tipo $a_1 = (v_1, v_2)$, $a_2 = (v_2, v_3)$, ..., $a_k = (v_k, v_{k+1})$ con partenza dal vertice v_1 e arrivo nel vertice v_{k+1} transitando per i vertici intermedi v_2, v_3, \dots, v_k . La presenza di un tale cammino fa sì che v_{k+1} sia raggiungibile a partire da v_1 ed introduce il concetto di raggiungibilità. Tale aspetto non è simmetrico, ma rappresenta l'estensione orientata del concetto di connessione visto precedentemente.

1.3. Rappresentazione dei grafi

Ci sono diversi modi per rappresentare un grafo e ciascuno di questi viene scelto in base al problema che si vuole risolvere, alle dimensioni del grafo e alle sue caratteristiche principali. I grafi, orientati e non orientati, devono poter essere rappresentati da strutture dati utilizzabili da un programma durante l'elaborazione al

calcolatore. Per esaudire tale necessità, tre possibili modi per rappresentare un grafo sono: *lista di adiacenza*, *matrice di adiacenza* e *matrice di incidenza* (o *matrice nodi – archi*).

1.3.1. Lista di adiacenza

Si rappresenta un grafo $G = (V, E)$ con un vettore Adj di liste, una lista per ogni vertice del grafo. Per ogni vertice u , $Adj[u]$ contiene tutti i vertici v adiacenti a u , ovvero quei vertici v tali per cui esiste un arco $(u, v) \in E$; in particolare questo insieme di vertici, secondo ordine arbitrario, è memorizzato come una lista. In figura 1.4 viene evidenziato l'impiego delle liste di adiacenza per visualizzare un grafo non orientato.

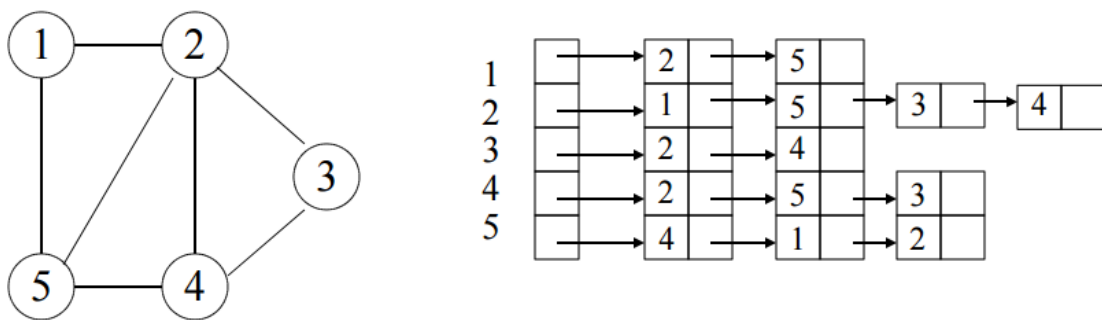


Figura 1.4 Visualizzazione di grafo non orientato mediante lista di adiacenza

In alcuni problemi si vuole poter associare una informazione (chiamata peso) ad ogni arco; in figura 1.5 viene mostrato l'impiego delle liste di adiacenza per visualizzare un grafo orientato pesato.

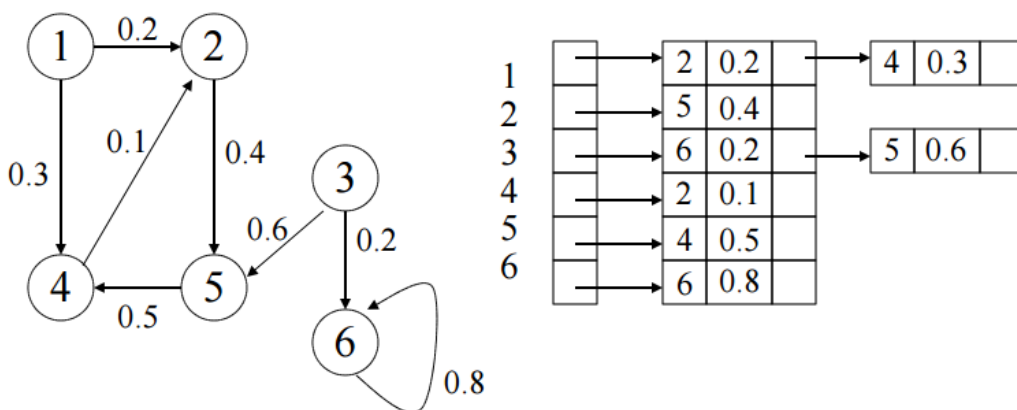


Figura 1.5 Visualizzazione di grafo orientato pesato mediante lista di adiacenza

La rappresentazione mediante lista di adiacenza viene preferita quando il grafo in esame è *sperso*, cioè con $|E|$ è molto minore di $|V|^2$.

1.3.2. Matrice di adiacenza

Per la rappresentazione con matrici di adiacenza si assume che i vertici siano numerati in sequenza da 1 a $|V|$. Si rappresenta un grafo $G = (V, E)$ con una matrice $Q = (q_{ij})$ di dimensioni $|V| \times |V|$ tale che:

$q_{ij} = 1$ se $(i, j) \in E$

$q_{ij} = 0$ altrimenti

La matrice di adiacenza di un grafo non orientato è simmetrica, ovvero $q_{ij} = q_{ji}$; in questo caso si possono considerare solo i dati sopra la diagonale (diagonale inclusa) riducendo di metà lo spazio necessario per la memorizzazione della matrice (vedi figura 1.6).

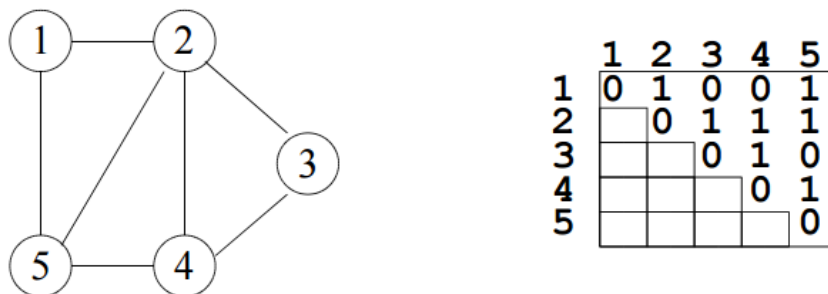


Figura 1.6 Visualizzazione di grafo non orientato mediante matrice di adiacenza

Di seguito, in figura 1.7, viene riportata la rappresentazione di un grafo orientato; in questo caso la matrice non risulta simmetrica e chiaramente tutti gli elementi al suo interno devono essere memorizzati in quanto portatori distinti di informazione.

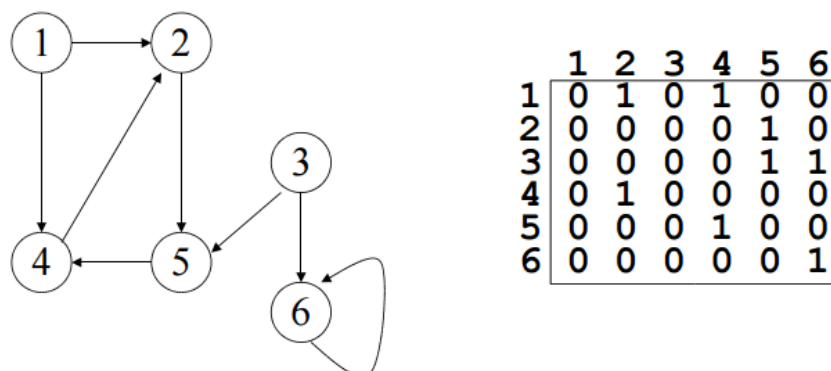


Figura 1.7 Visualizzazione di grafo orientato mediante matrice di adiacenza

La rappresentazione mediante matrice di adiacenza viene preferita quando il grafo in esame è *denso*, cioè con $|E|$ è molto vicino al valore di $|V|^2$.

1.3.3. Matrice di incidenza

Per la rappresentazione con matrici di incidenza (o matrice nodi – archi) si assume che i vertici siano numerati in sequenza da 1 a $|V|$ in modo arbitrario. Un grafo non orientato $G = (V, E)$ si rappresenta con una matrice $Q = (q_{ij})$ di dimensioni $|V| \times |E|$ tale che:

$$q_{ij} = + 1 \text{ se } e_j \text{ è incidente su } v_i$$

$$q_{ij} = 0 \text{ se } e_j \text{ non è incidente su } v_i$$

Nel caso di grafo orientato, l'elemento generico q_{ij} della matrice di incidenza deve tener conto della direzione dell'arco in esame. Un grafo orientato $G = (V, A)$ si rappresenta con una matrice $Q = (q_{ij})$ di dimensioni $|V| \times |A|$ tale che:

$$q_{ij} = + 1 \text{ se } a_j \text{ è arco entrante in } v_i$$

$$q_{ij} = - 1 \text{ se } a_j \text{ è arco uscente da } v_i$$

$$q_{ij} = 0 \text{ altrimenti}$$

Questa modalità di rappresentazione, visualizzata in figura 1.8, è basata su una matrice $n \times m$ in cui ogni riga corrisponde ad un vertice (nodo) ed ogni colonna è associata ad uno spigolo (arco). Nella matrice di incidenza ogni colonna ha quindi due elementi diversi da zero, in corrispondenza delle righe della matrice relative ai due nodi estremi dell'arco.

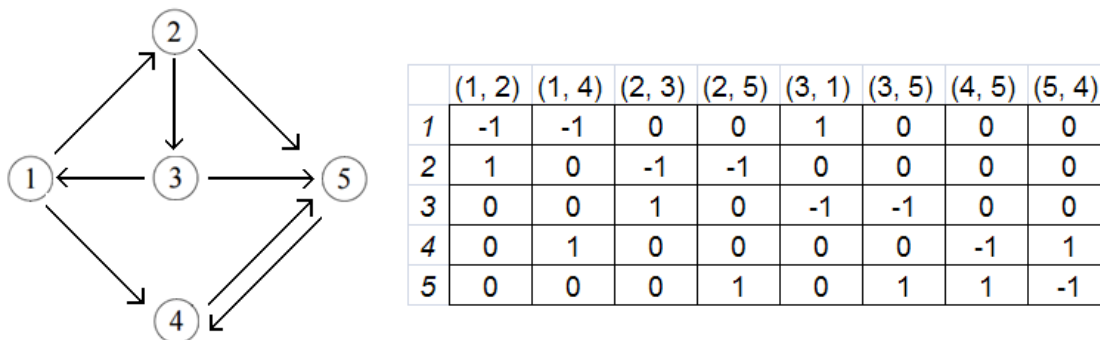


Figura 1.8 Visualizzazione di grafo orientato mediante matrice di incidenza

CAPITOLO 2

Vehicle Routing Problem (VRP)

Una flotta eterogenea di veicoli, in grado di fornire un servizio di raccolta e consegna di materiale ad un dato insieme di clienti con domande note, deve essere gestita in modo efficiente dal punto di vista economico per rappresentare un punto di forza ed un ruolo chiave nell'ambito dei trasporti.

In questo ambito esiste una classe di problemi logistici nota come *Vehicle Routing Problem (VRP)* le cui soluzioni mirano a minimizzare i costi totali (o le distanze) sostenuti dai veicoli durante l'assistenza ai clienti.

2.1. Definizione di VRP e concetti generali

Il *Vehicle Routing Problem*, comunemente abbreviato come *VRP*, è un nome generico dato ad un'intera classe di problemi che comportano la progettazione di percorsi (routes) ammissibili ottimi per una flotta di veicoli che deve servire una serie di clienti nel rispetto di determinati vincoli. Ogni cliente è servito una volta e tutti devono essere assegnati ai veicoli in modo tale da non superare la capacità massima del mezzo. L'aspetto centrale, dunque, è la pianificazione di questi percorsi su cui sono disposti i clienti da raggiungere e servire con l'obiettivo di minimizzare i costi di routing e di assegnamento dei veicoli.

Il problema logistico considerato può essere esplicitato andando a descrivere nel dettaglio le caratteristiche dei soggetti coinvolti che costituiscono elementi fondamentali per l'analisi del *VRP* stesso. Tali soggetti (*depositi, veicoli, rete stradale, clienti*) possono essere rappresentati graficamente per mezzo di grafi pesati nei quali i vertici identificano i clienti o depositi e gli archi raffigurano i diversi tratti stradali percorribili dai veicoli. Ad ogni arco viene attribuito un peso rappresentativo del costo di transito, del tempo o della distanza tra due nodi. In figura 2.1 viene riportato a titolo di esempio la soluzione di un *VRP* classico mediante un grafo pesato.

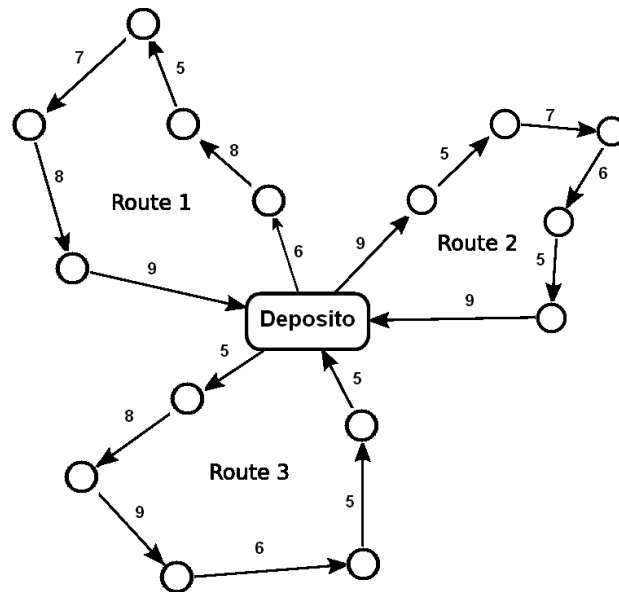


Figura 2.1 Soluzione di un VRP classico

2.1.1. Depositi

Il punto di partenza per lo studio del problema sono i *depositi* che presentano caratteristiche comuni o differenti a seconda delle ipotesi considerate:

- sono rappresentati graficamente come vertici del grafo;
- sono i punti di partenza per tutti i veicoli;
- possono essere molteplici oppure il deposito può essere unico, si può anche suddividere il problema per ciascun deposito;
- ad ogni deposito è associato un certo numero e tipologia di veicoli;
- ad ognuno di essi può essere legato un orario di lavoro entro il quale i mezzi devono uscire e rientrare al deposito assegnato;
- sono anche il punto di arrivo in quanto i veicoli alla fine del tragitto devono recarsi nuovamente al proprio deposito per una nuova assegnazione.

2.1.2. Veicoli

Ad ogni deposito è associato un certo quantitativo di *veicoli* con caratteristiche determinanti alla classificazione dei problemi di *VRP*:

- la flotta di veicoli associata ad un deposito può essere omogenea o eterogenea sia in termini di dimensioni che in termini di capacità;
- la capacità massima del mezzo dipende dal tipo di veicolo e dal tipo di saturazione considerata (questa può essere a peso, volume o numero di pezzi trasportati);

- ad ogni veicolo è associato un autista per il quale viene calcolato il costo del lavoro eseguito (costo di carico/scarico, costo orario, ecc.);
- altri costi associati al veicolo sono legati alla lunghezza del percorso assegnato sia in termini di spazio che in termini di tempo;
- ad ogni mezzo viene assegnato un percorso (l'insieme dei percorsi è un sottoinsieme della *rete stradale* che corrisponde all'insieme comprendente tutti gli archi del grafo).

2.1.3.Clienti

Lo studio del *VRP* è costruito attorno alle esigenze dei soggetti che devono ricevere il servizio, ossia i *clienti*:

- sono rappresentati dai vertici del grafo;
- ad ognuno di essi è associata una certa domanda che viene esplicitata come la quantità di merce da caricare e/o scaricare, che i veicoli (e pertanto gli autisti) si impegnano a portare correttamente ed efficientemente a destinazione;
- il servizio deve avvenire in determinati periodi di tempo legati agli orari di apertura e chiusura di ogni cliente;
- hanno differenti necessità di consegna/ritiro a seconda della merce trattata da ognuno di essi (ad esempio se è deperibile o meno o se necessita di essere trasportata e subire ulteriori lavorazioni fatte da altri soggetti);
- ogni cliente può essere servito da un unico veicolo che porta a termine l'intera richiesta oppure il servizio può essere effettuato da più mezzi (*split deliveries*), sempre dipendentemente dalle assunzioni fatte.

2.1.4.Obiettivi e Vincoli

Se ne deduce, pertanto, che il problema di trasporto delle merci implica un servizio che uno o più clienti richiedono venga soddisfatto, secondo specifiche esigenze, dai veicoli che sono localizzati in uno o più depositi ed effettuano i loro spostamenti attraverso la rete stradale "disponibile" e secondo le specifiche del singolo percorso ad essi assegnato. Il *VRP* permette di determinare un insieme di viaggi in modo da servire tutti i clienti nel rispetto dei vincoli operativi considerati atti a minimizzare la funzione di costo con l'obiettivo di minimizzare la totalità dei costi da considerare. I principali *obiettivi*, anche contrastanti, dei problemi di *VRP*, sono:

- ✓ minimizzare il numero di veicoli utilizzati per servire tutti i clienti;
- ✓ minimizzare la distanza totale percorsa dalla flotta;
- ✓ minimizzare il costo totale del trasporto che dipende dalla distanza totale percorsa, dal tempo totale impiegato e dai costi fissi associati ai veicoli;

- ✓ minimizzare le penali associate al servizio portato a termine solo a una parte dei clienti;
- ✓ bilanciare i percorsi per quanto riguarda il tempo di percorrenza e/o il carico del veicolo;
- ✓ minimizzare una funzione obiettivo che corrisponde ad una combinazione degli obiettivi precenti.

Tutti i soggetti coinvolti devono sottostare a determinati *vincoli*, senza i quali sarebbe impossibile effettuare una corretta analisi. Alcuni di essi sono comuni per le differenti tipologie di *VRP*, mentre altri sono specifici per ogni tipo di problema o sono il punto di partenza dello studio dai quali questo non può prescindere. Alcuni tipici vincoli sono i seguenti:

- la richiesta totale dei clienti posti lungo il percorso non può superare la capacità del veicolo ad esso assegnato;
- i clienti serviti possono richiedere solo la consegna di merce, solo la raccolta o ambedue le cose;
- i clienti possono essere serviti solo nei loro specifici intervalli temporali (*time windows*) e durante i periodi di lavoro degli autisti;
- devono essere rispettati eventuali vincoli di precedenza definiti tra i clienti. Si pensi, ad esempio, al caso in cui parte della merce da consegnare ad un cliente debba essere prima ritirata da altri (*pickup and delivery problem*); in questo caso, inoltre, interi gruppi di clienti devono essere serviti dallo stesso veicolo. Un'altra situazione di questo tipo si ha nel cosiddetto *VRP con Backhauls*, in cui i veicoli possono effettuare raccolta e distribuzione, a condizione che quest'ultima attività avvenga per prima.

Prima di passare alla descrizione delle diverse varianti di un problema di *VRP*, bisogna evidenziare il fatto che questo tipo di problema appartiene ad una classe di *problemi NP-hard*, vale a dire che il tempo necessario per risolverlo utilizzando algoritmi che ricercano la soluzione esatta cresce in maniera esponenziale con la dimensione del problema stesso.

2.2. Classificazione e varianti principali del VRP

Il *VRP* è un problema centrale per quanto concerne la distribuzione fisica di beni materiali e servizi. Esistono diverse tipologie di *VRP* a seconda della natura della merce trasportata, della qualità di servizio richiesta e delle caratteristiche di clienti e

veicoli. In tutti i casi, però, l'obiettivo rimane quello di soddisfare i clienti al costo minimo.

2.2.1. Capacitated Vehicle Routing Problem (CVRP)

Il *VRP* più semplice è il *Capacitated Vehicle Routing Problem (CVRP)*, che assume una flotta fissa di veicoli, con capacità uniforme, situata in un unico deposito centrale per soddisfare la domanda dei clienti. Ogni veicolo può essere utilizzato per un solo percorso e la domanda totale della rotta non può eccedere la capacità totale.

2.2.2. Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)

Il *Capacitated Vehicle Routing Problem with Time Windows* può essere considerato come una generalizzazione del *CVRP* con l'aggiunta, però, di vincoli quali il rispetto delle finestre temporali e altri dati di tipo temporale che comportano una maggiore complessità di analisi e studio. L'obiettivo di questi problemi è quello di minimizzare la flotta di veicoli e il tempo totale di viaggio e di attesa necessari per servire tutti i clienti nel rispetto dei loro vincoli temporali. L'intervallo di tempo (o *time window*) associato al cliente i viene indicato con $[e_i, l_i]$; il servizio di ogni cliente deve iniziare necessariamente in un istante t_i contenuto all'interno del *time window*. In caso di arrivo anticipato presso il cliente i , il veicolo deve attendere l'istante e_i (*earliest time*) prima di poter effettuare il servizio, generando un costo di attesa; in caso opposto, se il veicolo arriva dopo l'istante l_i (*latest time*), la soluzione risulta essere non fattibile. Ad ognuno dei clienti è associato un tempo di servizio s_i indicante appunto l'intervallo di tempo durante il quale il veicolo che effettua il servizio rimane fermo presso il cliente stesso. L'approfondimento di questa classe di problemi (*CVRPTW*) e la sua formulazione matematica vengono trattati in dettaglio nel terzo paragrafo di questo capitolo.

2.2.3. Distance-Constrained Vehicle Routing Problem (DCVRP)

La variante *CVRP* può essere modificata andando a sostituire il vincolo di capacità con un vincolo di lunghezza massima, o tempo massimo, del tragitto da percorrere (*DVRP*). In particolare, ad ogni arco viene associata una lunghezza non negativa, e la lunghezza totale, data dalla somma degli archi considerati, non può eccedere la lunghezza massima imposta come vincolo. Se i veicoli sono diversi tra loro, allora i valori massimi considerati sono diversi per ognuno di essi.

Se il parametro associato all'arco rappresenta il tempo di viaggio, ogni nodo i ha assegnato un tempo di servizio s_i che rappresenta il tempo necessario al mezzo per compiere il proprio servizio presso il cliente; in alcuni casi i tempi di servizio possono essere inclusi nei costi temporali degli archi, ponendo per ogni arco (i, j) :

$$t_{ij} = t'_{ij} + s_i / 2 + s_j / 2$$

dove t'_{ij} indica il costo temporale per la sola percorrenza dell'arco (i, j) .

Una variante del *DVRP* è data dal *Distance-Constrained Vehicle Routing Problem (DCVRP)*, in cui continuano a valere i vincoli di lunghezza o tempo di percorrenza massimo, ma viene mantenuto anche il vincolo di capacità di trasporto che prima veniva da questi sostituito. L'obiettivo del problema è minimizzare la lunghezza totale dei circuiti oppure, se il tempo di servizio è incluso nei costi temporali degli archi, la loro durata.

2.2.4. Vehicle Routing Problem with Backhauls (VRPB)

Questa variante è un'estensione del *CVRP* che prevede la divisione della totalità dei clienti in due sottoinsiemi:

- il primo, indicato con la lettera *L*, contiene clienti di tipo *Linehaul*, ossia clienti che devono ricevere una quantità di merce specificata dalla loro domanda;
- il secondo, indicato con la lettera *B*, contiene clienti *Backhaul*, ossia clienti presso i quali deve essere ritirata una certa quantità di merce.

Quando un mezzo serve entrambe le tipologie di clienti, viene imposto un vincolo di precedenza tra i due gruppi: tutti i clienti di tipo *L* devono essere serviti prima di ciascun cliente di tipo *B*.

Questa variante può anche essere modificata con l'inserimento di finestre temporali presso i clienti, e in tal caso prende il nome di *Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW)*.

2.2.5. Vehicle Routing Problem with Pickup and Delivery (VRPPD)

In questa tipologia di *VRP* viene esplicitato un vincolo di precedenza tra i clienti in quanto c'è la necessità di un certo tipo di servizio all'interno dello stesso viaggio. Ad ogni cliente i sono associati due parametri d_i (quantità di merce richiesta) e p_i (quantità di merce da ritirare); per non eccedere la capacità massima del veicolo si tiene conto della differenza netta $(d_i - p_i)$ necessaria al cliente.

La merce che viene richiesta da un certo vertice i deve essere prima raccolta presso il cliente O_i (nodo origine della domanda di consegna); allo stesso modo, la merce ritirata presso un determinato vertice i deve essere consegnata al cliente D_i (nodo destinazione merce) visitato dal veicolo subito dopo. Per convenzione si definisce che lo scarico della merce avviene prima del carico.

Spesso l'origine O_i e la destinazione D_i non sono indicati esplicitamente perché comuni a tutti i vertici (possono ad esempio coincidere con il deposito); in questo caso si tratta di un *Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD)*.

Alla tipologia di problemi *VRPPD* può essere associata un'ulteriore variante comprendente le finestre temporali per entrare nel campo dei *Vehicle Routing Problems with Pickup and Delivery and Time Windows (VRPPDTW)*.

Dopo la descrizione dettagliata e la classificazione delle varianti principali di un *VRP* segue in figura 2.2 lo schema logico riepilogativo.

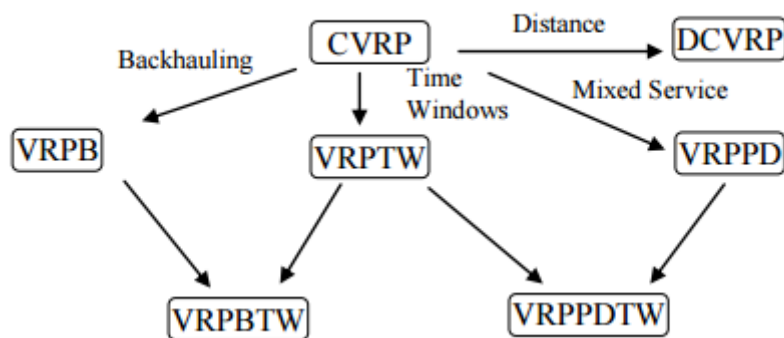


Figura 2.2 Varianti principali di un *VRP* e loro relazioni

Nel prossimo paragrafo viene descritta la versione *Capacitated Vehicle Routing Problem with Time Windows* utile per questo elaborato.

2.3. Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)

2.3.1. Aspetti generali del CVRPTW e vincoli

Il *Capacitated Vehicle Routing Problem with Time Windows* è, come visto, un'estensione del *CVRP* nella quale ad ogni cliente viene assegnato un intervallo di tempo $[e_i, l_i]$ chiamato appunto *finestra temporale*. Inoltre sono noti l'istante in cui il veicolo lascia il deposito, il tempo di viaggio ed il tempo impiegato dal mezzo per effettuare il servizio presso il cliente. Tale servizio deve avere inizio all'interno della finestra temporale associata ad ogni cliente, ed il mezzo necessita di rimanere presso il cliente per tutta la durata del servizio stesso. Inoltre, in caso di arrivo anticipato, il mezzo è costretto ad aspettare fino all'istante e_i , momento in cui può iniziare il servizio. Questo tempo d'attesa genera un costo che concorre alla composizione di una funzione di costo complessiva, la quale tiene conto in aggiunta di costi fissi (legati

all'uso dei mezzi) e costi operativi variabili (proporzionali al tempo di viaggio e al tempo di servizio).

L'obiettivo del *CVRPTW* è quello di cercare percorsi per ogni mezzo che minimizzino la funzione di costo complessiva e che rispettino i seguenti *vincoli*:

- ogni percorso inizia dal deposito;
- ogni cliente viene inserito in un solo percorso e ciò equivale a dire che può essere visitato da un solo veicolo;
- la capacità del veicolo assegnato al percorso deve essere superiore della somma delle domande dei clienti appartenenti al percorso stesso;
- il servizio deve iniziare in un istante interno alla finestra temporale $[e_i, l_i]$ associata al cliente per non generare costi d'attesa;
- il veicolo deve rimanere presso il cliente per tutto il tempo necessario al completamento del servizio (s_i);
- la durata del tempo totale durante il quale il veicolo può essere in servizio deve essere minore del tempo di lavoro massimo consentito tv_v^{max} .

Nei campi applicativi reali i vincoli di tempo e di capacità possono, in misura ridotta e controllata, essere violati in base alle caratteristiche delle finestre temporali, siano esse di tipo *soft* piuttosto che di tipo *hard*. Nel primo caso la soddisfazione della domanda del cliente può avvenire anche al di fuori delle predefinite finestre temporali, a patto che venga pagato un costo di penalizzazione (*penalty cost*). Il secondo caso, invece, impone il vincolo rigido che il servizio deve essere effettuato esclusivamente all'interno dei tempi prestabiliti dal cliente e, nel caso di arrivo anticipato del mezzo, viene generato il costo d'attesa come precedentemente indicato.

2.3.2. Definizione del problema e obiettivo

Il *CVRPTW* identifica una classe di problemi che può essere risolta attraverso modelli di ottimizzazione che, però, possono richiedere maggior tempo di risoluzione, oppure attraverso differenti metodologie euristiche che mirano a servire un certo numero di clienti utilizzando una combinazione di più veicoli disponibili. E' possibile fornire un modello matematico per il *CVRPTW* partendo dalla definizione del grafo che rappresenta il problema. Si considera infatti un grafo orientato $G = \{I, P, A\}$ nel quale i clienti rappresentati dai nodi $I = \{i_1, i_2, \dots, i_m\}$ e i depositi dai nodi $P = \{p_1, p_2, \dots, p_l\}$ sono connessi tra loro mediante archi rappresentati dall'insieme $A = \{(i, j) / i, j \in (I \cup P)\}$. Per ogni arco $(i, j) \in A$ vengono definiti un costo di trasporto c_{ij} e un tempo di percorrenza t_{ij} da i a j . Per ogni cliente si definisce una propria finestra temporale $[e_i, l_i]$, ed entro tale intervallo un carico fisso di merce w_i deve essere prelevato (o scaricato).

Per effettuare le operazioni di ritiro/consegna presso i clienti è disponibile una flotta eterogenea di veicoli $V = \{v_1, v_2, \dots, v_k\}$ con diverse capacità di carico indicate con q_v e alloggiata in più depositi $p \in P$. Ogni veicolo v deve partire dal deposito assegnato p , prelevare/scaricare la merce in più punti e tornare al deposito di partenza p . La rotta assegnata al veicolo v è definita come l'insieme dei nodi $r = (p, \dots, i, (i+1), \dots, p)$ connessi da archi orientati appartenenti ad A . Vengono successivamente definite due matrici per ogni mezzo, indicate con $C = \{c_{ij}^v\}$ e con $\Gamma = \{t_{ij}^v\}$, che riportano rispettivamente i costi di viaggio ed i tempi di viaggio che si generano dal nodo i al nodo j impiegando il veicolo v . Insieme alla domanda w_i di ogni cliente viene dato anche il tempo st_i^v impiegato dal mezzo per effettuare il servizio presso il determinato cliente.

L'*obiettivo* è quello di minimizzare il costo totale di carico e scarico nel servire tutti i nodi; devono essere considerati principalmente quattro tipologie di costo: costi fissi legati all'utilizzo del veicolo, costi di tempo e distanza nel percorrere i diversi tragitti, costi legati ai tempi d'attesa (quindi anche costi di penalità) e costi di lavoro.

CAPITOLO 3

Tecniche euristiche per la risoluzione di un VRP

Il termine *Euristica* indica un metodo (algoritmico) per la ricerca di soluzioni ammissibili (non necessariamente ottime) di un problema di ottimizzazione. I *CVRPTW* sono problemi abbastanza difficili dal punto di vista computazionale, infatti, se i *VRP* sono considerati *NP-hard* (dove *NP* sta per “polinomiale nel calcolo deterministico”) tanto più lo sono i *CVRPTW*. Per questa classe di problematiche risulta di fondamentale importanza lo sviluppo di tecniche euristiche aventi la finalità di trovare una soluzione ammissibile buona, vicina all’ottimo ed ottenuta in brevi tempi di elaborazione.

3.1. Classificazione tecniche euristiche

La costruzione di algoritmi euristici efficaci richiede un’attenta analisi del problema da risolvere volta ad individuare la “struttura”, ossia le caratteristiche specifiche utili, ed una buona conoscenza delle principali tecniche algoritmiche disponibili. Infatti, anche se ogni problema ha le sue caratteristiche specifiche, esistono tecniche generali che possono essere applicate, in modi diversi, a moltissimi problemi, producendo *classi* di algoritmi di ottimizzazione ben definite, esposte qui di seguito.

3.1.1. Euristici costruttivi

Gli algoritmi euristici costruttivi sono orientati all’individuazione (costruzione) di una soluzione ammissibile, trovata implementando un percorso alla volta attraverso una serie di step e contenendo il costo durante il procedimento. Possono essere schematizzati nelle tre fasi seguenti:

- I. *Inizializzazione*: per prima cosa si inizia il percorso cercando un elemento di partenza per la costruzione della soluzione parziale *S*.
- II. *Selezione di un nuovo elemento da aggiungere alla soluzione parziale*: viene individuato il criterio di selezione di un nuovo elemento da aggiungere alla soluzione parziale *S*.
- III. *Criterio di arresto*: se *S* è completa, e quindi ammissibile, la procedura viene interrotta, altrimenti si ritorna al passo precedente

A loro volta, gli algoritmi euristici costruttivi possono essere divisi in due categorie:

- ✓ Sequenziali: costruiscono un percorso alla volta fino a quando tutti i clienti sono soddisfatti.
- ✓ Paralleli: costruiscono più rotte contemporaneamente. Il numero di percorsi può essere fissato a priori oppure derivare dalla fusione di percorsi minori.

3.1.2.Euristici a due fasi

I metodi a due fasi scompongono il problema nelle operazioni di suddivisione dei vertici in gruppi (*cluster*) e di costruzione di *route* ammissibili. A loro volta, gli algoritmi a due fasi possono essere divisi in due classi:

- ✓ Cluster first – Route second: i vertici sono inizialmente raggruppati in cluster ed in un secondo momento viene costruito un route per ogni cluster.
- ✓ Route first – Cluster second: un route viene costruito su tutti i vertici per poi essere suddiviso.

3.1.3.Euristici migliorativi

I metodi euristici migliorativi si applicano ad una soluzione preesistente (in alcuni casi anche non ammissibile) con l'intento di migliorarla e operano tipicamente mediante scambio di archi o vertici tra diversi percorsi

3.2.Tipologie di tecniche euristiche

Di seguito vengono analizzate e spiegate alcune delle principali tecniche euristiche (Solomon, 1987).

3.2.1.Savings Heuristics

Questa tecnica è di tipo costruttivo – parallelo ed inizia con n percorsi distinti in ognuno dei quali il cliente è servito da un veicolo dedicato; ad ogni iterazione l'obiettivo è quello di aggregare due o più nodi tra loro, ma anche due o più percorsi con più di un cliente, e calcolarne il risparmio (*saving*) derivante da queste operazioni. Dati due clienti, serviti ciascuno da un mezzo dedicato con rotta distinta, l'applicazione del saving mette in evidenza il potenziale risparmio che si può avere impiegando un unico veicolo che visita prima un cliente e poi il successivo, percorrendo un unico tragitto. Avendo indicato con i e j due clienti generici e con O il deposito del mezzo, basta applicare la formula seguente per il calcolo del risparmio:

$$sav_{ij} = d_{iO} + d_{Oj} - \mu d_{ij}, \text{ con } \mu \geq 0$$

Questo risparmio viene calcolato considerando le distanze dal punto di vista spaziale. Per esempio, considerando $\mu = 1$, si ottiene il risparmio in termini di distanza (o costo)

che risulta dal servire i clienti i e j attraverso un unico percorso, piuttosto che servirli individualmente direttamente dal deposito. Nel *CVRPTW* si deve però considerare l'orientamento dei percorsi in quanto i clienti devono essere vicini anche dal punto di vista temporale. Ad ogni iterazione, oltre a tener conto del vincolo di capacità totale del veicolo, è necessario controllare che non vengano violati i vincoli legati alle finestre temporali. Può infatti risultare vantaggioso unire due clienti molto vicini in termini di spazio, ma rivelarsi svantaggioso o addirittura penalizzante in termini di tempo. Possono generarsi dei tempi di attesa che vanno a creare dei costi opportunità: il veicolo, ad esempio, invece di attendere fino all'istante di apertura di un determinato cliente, nel frattempo ha la possibilità di servirne altri. Per la soddisfazione della vicinanza spazio – temporale tra due clienti si cerca di limitare il tempo di attesa quando si uniscono due clienti; per far ciò si usa un parametro che esclude l'unione dei due clienti nel caso in cui il tempo di attesa risulta essere maggiore del parametro stesso. L'algoritmo si arresta e termina quando la domanda dei clienti è soddisfatta dai veicoli che percorrono le rotte costruite.

3.2.2.A Time-Oriented, Nearest-Neighbor Heuristic

Questa tecnica di tipo costruttivo – sequenziale si basa sull'inizializzazione di una rotta cercando il cliente più vicino al deposito che non è ancora stato inserito in nessun percorso. Ad ogni successiva iterazione viene aggiunto nel percorso il cliente (non ancora assegnato) più vicino all'ultimo inserito nel percorso stesso. Questo nuovo cliente deve rispettare le finestre temporali, l'orario di ritorno del veicolo al deposito ed i vincoli di capacità del mezzo; quando nessuno dei successivi clienti rispetta questi vincoli imposti, si inizia allora una nuova rotta. La metrica usata in questo tipo di approccio tenta di soddisfare sia la vicinanza temporale che la vicinanza spaziale; infatti se si aggregano due clienti, si calcola sia la distanza (d_{ij}) che la differenza di tempo tra la fine del servizio in un nodo e l'inizio dello stesso nel successivo (T_{ij}). Inoltre si tiene conto anche dell'urgenza di servire un determinato cliente, espressa come il tempo rimanente prima che l'ultimo servizio del mezzo abbia inizio (v_{ij}). Indicando con s_i il tempo di servizio presso il cliente i , con t_{ij} il tempo di viaggio da i a j , con $[e_j, l_j]$ la finestra temporale associata al cliente j , si ha che:

$$T_{ij} = e_j - (e_i + s_i)$$

$$v_{ij} = l_j - (e_i + s_i + t_{ij})$$

$$c_{ij} = \delta_1 d_{ij} + \delta_2 T_{ij} + \delta_3 v_{ij}$$

$$\text{con } \delta_1 + \delta_2 + \delta_3 = 1$$

$$\text{e con } \delta_1, \delta_2, \delta_3 \geq 0$$

3.2.3. Insertion Heuristics

Gli algoritmi di inserzione sono considerati algoritmi costruttivi – sequenziali; la fase di inizializzazione, ossia la scelta del cliente da inserire nel tragitto come primo nodo, avviene seguendo uno dei tre criteri seguenti:

- i. cliente con distanza maggiore dal deposito;
- ii. cliente con orario di chiusura (della propria finestra temporale) minore
- iii. cliente con costo minore in termini di tempo e distanza dal deposito.

Dopo aver inizializzato una nuova rotta, si utilizzano due criteri, $c_1(i, u, j)$ e $c_2(i, u, j)$, per inserire ad ogni iterazione un cliente u all'interno della rotta corrente tra due nodi adiacenti i e j . Sia $(i_0, i_1, i_2, \dots, i_m)$ la rotta che si sta considerando, con $i_0 = i_m = 0$. Per ogni cliente u non ancora assegnato, quindi fuori dal percorso, si calcola il miglior inserimento nella rotta in considerazione:

$$c_1(i(u), u, j(u)) = \min [c_1(i_{p-1}, u, i_p)] \quad \text{con } p = 1, \dots, m$$

Inserendo il cliente u tra i_{p-1} e i_p si possono alterare tutti i tempi di inizio del servizio per i nodi della rotta $(i_0, i_1, i_2, \dots, i_m)$. Il cliente successivo da inserire sarà quello per cui vale la seguente relazione:

$$c_2(i(u^*), u^*, j(u^*)) = \text{optimum} [c_2(i(u), u, j(u))] \quad \text{dove } u \text{ è senza rotta e fattibile.}$$

Il cliente u^* viene quindi inserito nel percorso tra $i(u^*)$ e $j(u^*)$. Questa iterazione continua fino a quando non è più possibile trovare clienti che rispettano i vincoli e pertanto il percorso considerato termina; se non sono ancora stati soddisfatti tutti i clienti si procede con l'inizializzazione di una nuova rotta.

Ora vengono considerati tre diversi approcci specifici, basati sul criterio generale appena presentato.

3.2.3.1. Insertion Heuristic I1

Il primo criterio cerca di massimizzare il beneficio derivante dal servire un cliente presente nella rotta parziale in fase di costruzione piuttosto che dover servire il cliente con una rotta diretta.

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij} \quad \text{con } \mu \geq 0$$

$$c_{12}(i, u, j) = e_{ju} - e_j$$

La variabile e_{ju} sta ad indicare il nuovo istante in cui il servizio presso il cliente j può avere inizio dal momento che nel percorso è stato aggiunto il nodo u .

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) \quad \text{con } \alpha_1 + \alpha_2 = 1$$

$$\text{e con } \alpha_1, \alpha_2 \geq 0$$

$$c_2(i, u, j) = \lambda d_{0u} - c_1(i, u, j) \quad \text{con } \lambda \geq 0$$

Con questo tipo di metodo si massimizza il beneficio derivante dall'inserimento del cliente all'interno della rotta parziale, piuttosto che servito mediante rotta diretta. Il miglior inserimento è quello che minimizza una combinazione pesata di distanza e tempo, ossia che minimizza una misura di extra distanza ed extra tempo richiesti per visitare un cliente. Chiaramente diversi valori di μ e λ portano a diversi criteri per selezionare l'inserimento di un nodo nel percorso.

3.2.3.2. Insertion Heuristic I2

Il secondo tipo di criterio seleziona i clienti da inserire nel tragitto con l'obiettivo di minimizzare i costi sia in termini di tempo che di distanza totali.

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) \quad \text{con } \alpha_1 + \alpha_2 = 1$$

$$\text{e con } \alpha_1, \alpha_2 \geq 0$$

$$c_2(i, u, j) = \beta_1 R_d(u) - \beta_2 R_t(u) \quad \text{con } \beta_1 + \beta_2 = 1$$

$$\text{e con } \beta_1, \beta_2 \geq 0$$

Con $R_d(u)$ e con $R_t(u)$ si indicano rispettivamente la distanza totale e il tempo totale del tragitto parziale che si generano con il cliente u inserito.

3.2.3.3. Insertion Heuristic I3

Nel terzo approccio viene tenuta in considerazione, all'interno dell'aspetto temporale, anche l'urgenza di dover servire un cliente.

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij} \quad \text{con } \mu \geq 0$$

$$c_{12}(i, u, j) = e_{ju} - e_j$$

$$c_{13}(i, u, j) = l_u - e_u$$

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) + \alpha_3 c_{13}(i, u, j) \quad \text{con } \alpha_1 + \alpha_2 + \alpha_3 = 1$$

$$\text{e con } \alpha_1, \alpha_2, \alpha_3 \geq 0$$

$$c_2(i, u, j) = c_1(i, u, j)$$

Si può notare che questo tipo di euristiche sono una generalizzazione del *Time-Oriented, Nearest-Neighbor Heuristics*; infatti, invece di inserire un cliente alla fine della rotta, si inserisce in una posizione fattibile compresa tra una coppia di clienti già presenti nella rotta stessa. In tutti gli approcci presentati l'inserimento di un cliente in una rotta è guidato e condizionato sia da criteri geografici che da criteri temporali. Ci si aspetta perciò che il tempo d'attesa nelle soluzioni proposte da questo algoritmo euristico sia significativamente più basso rispetto a quello proposto dai criteri basati sulle sole distanze.

3.2.4.A Time-Oriented Sweep Heuristic

Questa euristica scompone il *Capacitated Vehicle Routing Problem with Time Windows* in una fase di raggruppamento (*clustering*) ed in una fase di pianificazione (*scheduling*). Nella prima fase si assegnano i veicoli ai clienti, mentre nella seconda si costruisce una pianificazione del veicolo usando un'euristica di tipo costruttivo, per soddisfare i clienti di un determinato settore.

Per rispettare la vicinanza geografica, si possono usare differenti criteri di selezione per la creazione del gruppo successivo (*cluster*). Si usa una semplice regola che divide in due il settore appena considerato e inserisce i clienti a seconda del loro angolo polare rispetto al centro considerato. (si assume il senso antiorario e si considera il più piccolo angolo formato dalla bisettrice e dalla retta che va dal deposito al cliente). Il percorso viene ripetuto fino a quando tutti i clienti sono stati inseriti all'interno di un percorso.

Nello specifico la tecnica euristica *Cluster* unita alla tecnica costruttiva – sequenziale di inserzione (*Cheapest Insertion*) sono quelle sviluppate all'interno di questo elaborato per risolvere un *CVRPTW*.

CAPITOLO 4

CVRPTW: raccolta, trasporto e consegna delle pelli

Dopo la presentazione, avvenuta nei capitoli precedenti, dei concetti teorici di base, delle varie tipologie di problemi ed alcune metodologie per la loro risoluzione, segue ora una descrizione dettagliata del problema oggetto di questa tesi. Lo studio riguarda il *CVRPTW* legato al trasporto di pelli bovine fresche da dei macelli (dove sono state lavorate) ad una conceria; il problema considerato è uno specifico caso di routing con vincoli certi dai quali la soluzione non può prescindere: come già accennato, ci sono specifiche finestre temporali da rispettare e la capacità dei mezzi disponibili al trasporto merci è limitata. L'obiettivo di questo elaborato, dopo attenta analisi del *CVRPTW*, è quello di sviluppare e costruire un algoritmo in *VBA* per la soluzione e l'organizzazione del problema di routing in questione. In questo capitolo vengono descritti il problema e l'ambiente oggetto di studio, i dati, le variabili ed i vincoli che lo caratterizzano.

4.1.Descrizione del problema

Lo studio riguarda un *Capacitated Vehicle Routing Problem with Time Windows* di trasporto di pelli bovine fresche da dei macelli ad una conceria; il problema è caratterizzato dalla presenza di *finestre temporali* per ogni macello (o conceria) presso il quale si raccolgono (o consegnano) le pelli e da *vincoli di capacità* sui veicoli impiegati. Chiaramente la questione è di grande utilità, in quanto può essere adattata ad altri problemi simili che si possono presentare ad analisi future con medesima struttura di vincoli ed impostazioni.

L'ambiente del problema può essere visto come un territorio con una rete stradale lungo la quale sono collocati, in posizioni fisse, una conceria, un certo numero di macelli per la lavorazione di pelli e uno o più depositi per i veicoli adibiti al trasporto delle merci. I macelli forniscono pelli bovine fresche che devono essere trasportate alla conceria con l'ausilio di camion eterogenei, nel rispetto dei vincoli imposti dalle finestre temporali e dalla capacità limitata del camion stesso; i veicoli, parcheggiati presso i depositi, sono caratterizzati ognuno da una capacità massima di trasporto, espressa in termini di peso. Ogni mezzo, se utilizzato, inizia il tragitto dal proprio deposito e

comporta un costo fisso di uscita, al quale può sommarsi un'altra componente di costo nel caso in cui il veicolo ritardi la propria partenza. Ogni macello fornisce una determinata quantità di merce e può essere servito da un unico mezzo; l'accoppiata di questi soggetti è contraddistinta da specifici tempi e costi di carico. I tempi ed i costi di viaggio, legati allo spostamento da un macello al successivo, possono variare e dipendono dal mezzo impiegato; quest'ultimo, una volta completato il tragitto assegnato, deve effettuare la consegna della merce raccolta alla conceria rispettando la finestra temporale della stessa. Inoltre, vista la deperibilità delle merci, ciascun macello impone un tempo massimo di scarico alle pelli che devono essere trasportate dallo stesso alla conceria. Il peso totale delle pelli caricate da ogni veicolo non può ovviamente superare quello massimo consentito dal mezzo stesso (vincolo di capacità).

L'obiettivo consiste nel trovare i tragitti che ciascun camion candidato alla partenza deve seguire per raccogliere le merci (pelli fresche), fornite da tutti i clienti (macelli) nel rispetto dei vincoli di capacità e finestra temporale imposti, e per minimizzare il costo totale, dato dal costo d'uscita sommato al costo di viaggio, al costo di carico/scarico, al costo del tempo impiegato ed eventualmente al costo d'attesa.

4.1.1. Soggetti che contraddistinguono il problema

Il caso oggetto di studio può essere rappresentato mediante grafo orientato (come si vede in figura 4.1); i soggetti considerati sono principalmente di quattro tipi: il deposito (o i depositi) in cui risiedono i mezzi, i macelli, la conceria e la rete stradale che li collega. I primi tre sono rappresentati nel grafo come nodi mentre le tratte che compongono la rete stradale (e quindi i possibili percorsi assegnati ai mezzi) sono rappresentate da archi pesati.

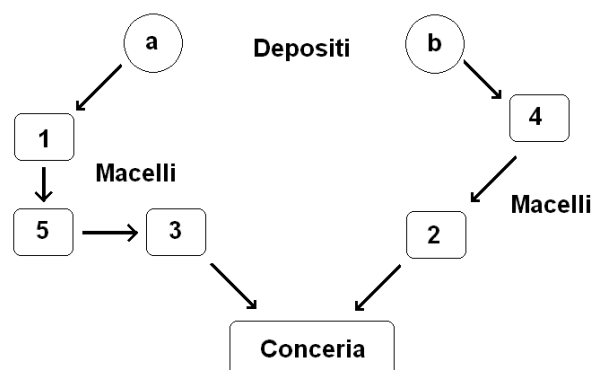


Figura 4.1 Rappresentazione grafica del CVRPTW

4.1.1.1. Deposito

Il punto di partenza per lo studio è il *deposito* (o i depositi) in quanto sede di tutti i *camion* utilizzati per il trasporto delle merci. Dal punto di vista geografico il deposito rappresenta la partenza effettiva, infatti, i camion, una volta effettuato l'intero tragitto a loro assegnato, devono rientrare nel deposito stesso per ricevere il successivo incarico. I veicoli a disposizione sono di tipo eterogeneo, diversi quindi tra loro e con capacità di carico distinte. Nel caso considerato la saturazione del mezzo avviene "a peso" e ciò rende possibile il caricamento del veicolo per un peso complessivo di merce inferiore o uguale a quello caratterizzante la capacità del mezzo stesso.

4.1.1.2. Macelli

I mezzi visitano i clienti (*macelli*), disposti geograficamente in punti distinti, per caricare la quantità di merce (pelli) specifica per ognuno di essi. Le pelli fresche fornite dai macelli necessitano di essere lavorate entro tempi brevi (ore), altrimenti possono subire alterazioni e deperimento irreparabile con conseguente danno economico. A tal proposito diventa fondamentale l'organizzazione dei tragitti che devono assolutamente tener conto della produzione quotidiana dei macelli, dei loro tempi di apertura e chiusura (entro i quali deve avvenire il ritiro della merce) e del tempo massimo di scarico merce (entro il quale le pelli devono raggiungere la conceria per essere lavorate ed evitare il deperimento).

4.1.1.3. Conceria

La *conceria* può essere considerata il punto finale del tragitto, o meglio, il penultimo nodo che il mezzo visita prima di far ritorno al deposito di partenza. In questo caso studio viene considerato fisso il costo di rientro del camion, per la tratta conceria deposito, che quindi non va ad influire nella ricerca del costo minimo tra i percorsi considerati.

4.1.1.4. Rete stradale

Come già anticipato, ad ogni mezzo è associato un tragitto che comporta la visita di più nodi nel rispetto dei vincoli di tempo e di capacità. La *rete stradale* è l'insieme di tutti i percorsi che i camion possono effettuare; viene rappresentata dagli archi del grafo in questione, come illustrato in figura 4.1.

4.2. Dati, indici e notazioni specifiche del problema

Per focalizzare il problema è fondamentale attribuire una notazione a ciascuno dei soggetti coinvolti, in particolare:

- ✓ i camion, la cui quantità viene indicata con r , sono indicizzati con k ;
- ✓ ogni mezzo k è associato ad un deposito m_k ;
- ✓ sono presenti n macelli identificati con i, j ;
- ✓ una conceria è indicata con l .

Ad ogni mezzo k è abbinato uno dei seguenti dati:

- q_k , capacità;
- c_k , costo fisso di uscita generato ogni volta che il mezzo lascia il proprio deposito per effettuare un servizio;
- ct_k , costo unitario di tempo da considerare per l'intero periodo durante il quale il veicolo è in servizio;
- cr_k , costo di attesa nell'unità di tempo da tenere in considerazione ogni volta che il mezzo ritarda l'uscita dal deposito, rispetto all'orario di apertura indicato nella sua finestra temporale;
- $[sm_k, fm_k]$, finestra temporale del deposito m_k e quindi associata al mezzo k .

Ad ogni macello i (o conceria l) è abbinato uno dei seguenti dati:

- p_i , il peso totale delle pelli fresche da caricare nel macello i ;
- $[s_i, f_i]$, la finestra temporale di carico;
- g_i , tempo massimo di scarico presso la conceria l delle pelli caricate nel macello i per evitare il deterioramento della merce;
- $[s_l, f_l]$, la finestra temporale della conceria.

Infine per ogni associazione camion – macello, camion – macello – macello e camion – macello – conceria si hanno i seguenti dati:

- d_{mik}, t_{mik} , costo e tempo di spostamento dal deposito m al macello i per ogni camion k ;
- dt_{ik}, tt_{ik} , costo e tempo di carico per ogni macello i per ogni camion k ;
- d_{ijk}, t_{ijk} , costo e tempo di spostamento dal macello i al macello j per ogni camion k ;
- d_{jlk}, t_{jlk} , costo e tempo di spostamento dal macello j alla conceria l per ogni camion k ;

- dt_{lk} , tt_{lk} , costo e tempo di scarico delle pelli alla conceria l per ogni camion k .

In figura 4.2 viene riportata la rappresentazione grafica dei dati del problema, per ogni camion k .

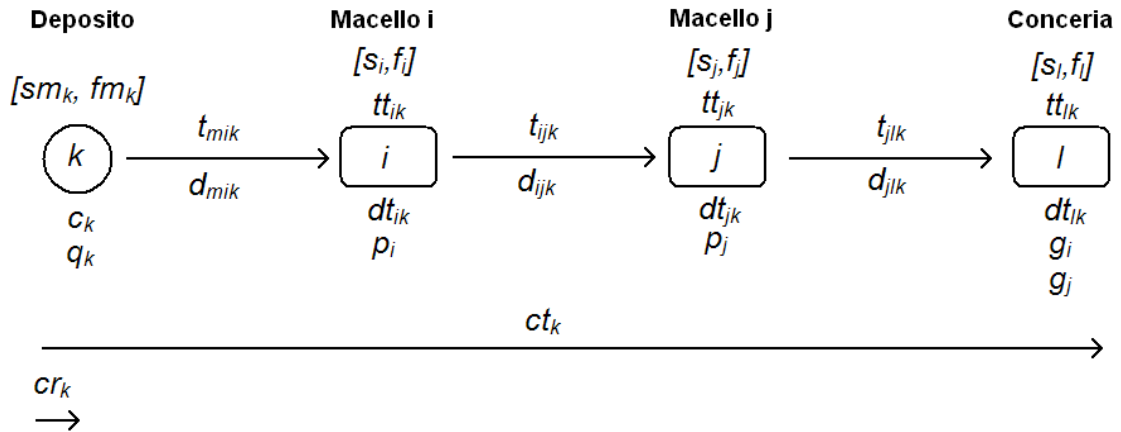


Figura 4.2 Rappresentazione grafica dei dati del CVRPTW

4.3. Funzione obiettivo e vincoli del problema

4.3.1. Funzione obiettivo

La funzione obiettivo del *Capacitated Vehicle Routing Problem with Time Windows* tende a minimizzare il costo totale generato da ogni mezzo durante l'esecuzione del tragitto assegnato; questo costo totale, per ogni camion k , è dato dalla sommatoria delle seguenti singole voci:

- ✓ costo di uscita del mezzo;
- ✓ costo totale di viaggio;
- ✓ costo totale di carico
- ✓ costo totale di scarico;
- ✓ costo totale del tempo impiegato per il servizio completo;
- ✓ costo dell'eventuale partenza ritardata del veicolo dal deposito (*start delay*).

4.3.2. Vincoli del problema

Per poter risolvere correttamente il CVRPTW oggetto di studio, bisogna tenere in considerazione i vincoli che, per semplicità, possono essere divisi in tre gruppi come di seguito esposto.

- Vincoli di viaggio:** impongono, ad esempio, che ogni macello debba essere servito da un unico mezzo che ha la responsabilità di soddisfare completamente la domanda del cliente. Devono inoltre essere rispettate le

precedenze tra i nodi a seconda del percorso assegnato al veicolo (infatti vi è un certo “sequenziamento” deciso nell’attribuzione dei clienti da visitare). Infine ogni mezzo deve terminare il proprio percorso alla conceria.

- ii. **Vincoli di tempo**: si riferiscono al rispetto delle finestre temporali, ossia il periodo di tempo, proprio di ogni soggetto coinvolto, durante il quale è possibile effettuare il servizio.
- Per i depositi la finestra temporale indica il periodo di tempo all’interno del quale i mezzi effettuano l’uscita per servire i clienti ed entro il quale devono effettuare la partenza.
 - Per i macelli la finestra temporale definisce l’orario di apertura (istante dal quale il mezzo può entrare per il carico della merce) e l’orario di chiusura (istante oltre il quale non è più possibile effettuare il servizio). Nel caso di arrivo anticipato da parte del veicolo, viene generato un costo che tiene conto, in termini di costo orario di lavoro, del tempo d’attesa per l’apertura del macello.
 - Per la conceria la finestra temporale definisce l’orario di apertura (istante dal quale il mezzo può entrare per lo scarico della merce) e l’orario di chiusura (istante oltre il quale non è più possibile effettuare il servizio).

L’ultimo vincolo di tempo è dato dai tempi massimi che ogni macello impone per lo scarico in conceria della propria merce, al fine di preservare le pelli fresche e non incorrere in deperimento e perdite economiche.

- iii. **Vincoli di capacità del camion**: si riferiscono semplicemente al fatto che la quantità di merce caricata da un mezzo (data dalla somma delle quantità di ogni cliente servito nel suo percorso) non può superare la sua capacità di carico. In questo problema, come già anticipato, la capacità è calcolata “a peso” e quindi espressa in chilogrammi (Kg).

4.3.3. Variabili del problema

Le variabili di interesse che contraddistinguono il *CVRPTW* sono:

- i veicoli in partenza;
- il tragitto seguito da ogni mezzo che definisce l’ordine con cui vengono serviti i clienti;
- l’istante in cui ogni veicolo inizia a servire ogni cliente.

La soluzione del problema di routing considerato ha come finalità l'organizzazione di una rete di trasporto, impiegando veicoli eterogenei con capacità limitate al servizio di soggetti contraddistinti da specifiche finestre temporali. Questa soluzione, considerata buona e ammissibile, viene trovata mediante l'impiego della tecnica euristica *Cluster* unita alla tecnica costruttiva – sequenziale di inserzione (*Cheapest Insertion*), spiegate dettagliatamente nei prossimi capitoli.

CAPITOLO 5

L'algoritmo euristico Cluster per la soluzione del CVRPTW

Come già riportato precedentemente, la soluzione del Vehicle Routing Problem with Time Windows, aspetto centrale di questa tesi, viene ottenuta applicando un metodo euristico a due fasi di tipo *Cluster first – Route second*. Nella prima fase il metodo scompone il problema nelle operazioni di suddivisione dei vertici in gruppi (*cluster*) e nella seconda costruisce i *route* ammissibili.

In questo capitolo vengono presentate le basi teoriche dell'algoritmo euristico cluster, utili al raggruppamento dei clienti e loro assegnazione ai veicoli; la spiegazione della costruzione dei *route* ammissibili attraverso l'euristica costruttiva – sequenziale di *Cheapest Insertion* viene riportata nel capitolo successivo.

5.1.CVRPTW con depositi multipli e veicoli eterogenei

5.1.1.Definizione del problema

Si considera una rete di percorsi, rappresentata dal grafo orientato $G = \{I, P, A\}$, che collega i nodi dei clienti (macelli) $I = \{i_1, i_2, \dots, i_n\}$ e i nodi dei depositi $P = \{p_1, p_2, \dots, p_d\}$ attraverso un insieme di archi orientati $A = \{(i, j) / i, j \in (I \cup P)\}$. L'arco $(i, j) \in A$ dovrebbe essere il tragitto con costo più basso di percorrenza che collega il nodo i al nodo j . I clienti i chiedono la raccolta/consegna di un carico fisso w_i all'interno della finestra temporale $[s_i, f_i]$, dove s_i indica l'orario di apertura e f_i quello di chiusura. Una flotta di veicoli eterogenei $V = \{v_1, v_2, \dots, v_m\}$, con diverse capacità di carico (q_v) e alloggiati in depositi $p \in P$, è a disposizione per svolgere i compiti richiesti di carico/scarico. Ogni veicolo v deve lasciare il deposito assegnato p , prendere il carico w_i presso i clienti i e alla fine del tragitto assegnato far ritorno al punto di partenza p . Il percorso per ogni veicolo v è un tour di nodi $r = (p, \dots, i, (i+1), \dots, p)$, collegati da archi orientati che appartengono ad A , che inizia e termina nel deposito p assegnato al veicolo v . Gli elementi di A sono archi orientati pesati, infatti, ad ogni elemento $a_{ij} \in A$ sono associati due valori: c_{ij} (costo di viaggio dal nodo i al nodo j usando il veicolo v) e t_{ij} (tempo di viaggio dal nodo i al nodo j usando il veicolo v). Si assume che c_{ij} e t_{ij}

soddisfino rispettivamente le disuguaglianze triangolari $c_{ik} + c_{kj} \geq c_{ij}$ e $t_{ik} + t_{kj} \geq t_{ij}$. Sono noti per ciascun cliente i la domanda w_i ed il tempo di servizio del veicolo v (st_i^v).

Una soluzione fattibile del *CVRPTW* deve soddisfare i seguenti *vincoli*:

- ogni percorso deve iniziare e finire presso lo stesso deposito;
- ogni nodo deve essere servito solo da un singolo veicolo;
- il carico totale assegnato al veicolo v non deve superare la sua capacità di carico q_v ;
- il periodo di tempo durante il quale un veicolo v può essere in servizio deve essere inferiore al tempo di lavoro massimo consentito $t_{V_v}^{max}$;
- il servizio di carico/scarico presso ogni sede dei clienti i deve avvenire all'interno della finestra temporale $[s_i, f_i]$, altrimenti deve essere addebitato un costo di penalità.

L'*obiettivo* del problema è di minimizzare il costo totale per l'esecuzione dei servizi di raccolta (o consegna) presso tutti i nodi dei clienti. Nella funzione obiettivo sono considerati quattro tipi di costi: i costi fissi per i mezzi utilizzati, i costi variabili legati alla distanza ed al tempo di viaggio, i costi del tempo d'attesa ed i costi di penalità dovuti alle violazioni dei vincoli del tempo di lavoro e della finestra temporale.

In figura 5.1 viene rappresentato graficamente il problema appena esposto, oggetto di studio in questo elaborato.

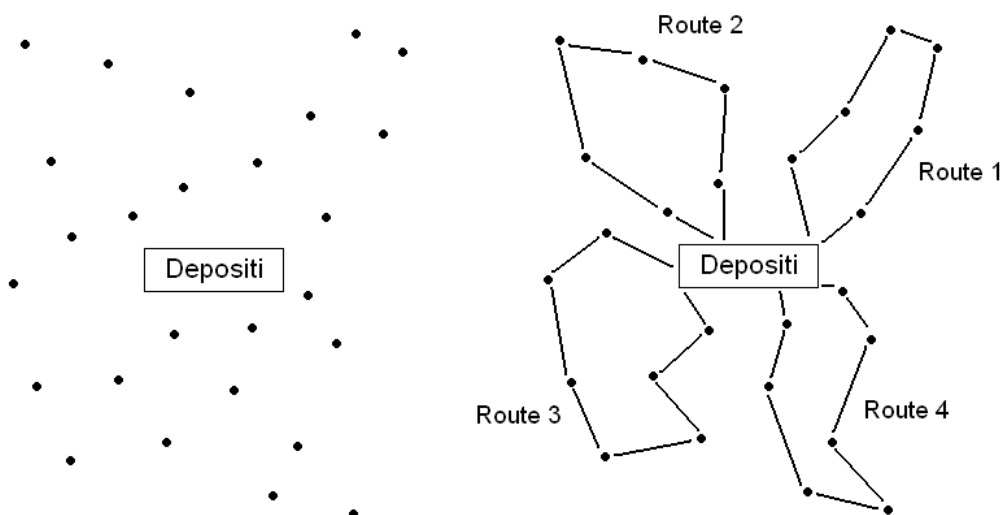


Figura 5.1 Rappresentazione grafica del modello

5.1.2. Variabili decisionali e formulazione matematica del problema

La formulazione matematica del problema richiede di definire tre diversi insiemi di variabili binarie:

- la variabile di assegnazione Y_{iv} per assegnare il mezzo $v \in V$ all'ubicazione del cliente $i \in I$;
- la variabile di assegnazione X_{pv} per utilizzare il mezzo $v \in V$ situato ne deposito $p \in P$;
- la variabile di sequenziamento S_{ij} per indicare che l'ubicazione del cliente $i \in I$ è visitata prima ($S_{ij} = 1$) o dopo ($S_{ij} = 0$) del nodo del cliente $j \in I$; questo è valido se i e j sono serviti entrambi dallo stesso veicolo v altrimenti S_{ij} perde di significato.

Per ogni coppia di nodi (i, j) , che può prendere parte allo stesso percorso, viene definita solo una singola variabile S_{ij} che assume il valore unitario per ogni nodo i visitato prima del nodo j dallo stesso veicolo v .

La formulazione matematica del CVRPTW con flotta di veicoli eterogenei e depositi multipli è la seguente funzione obiettivo:

$$\text{Min} \left(\sum_{v \in V} \left(\left(c_{f_v} \sum_{p \in P} x_{pv} \right) + (c_t TV_v) + (CV_v) + (\rho_v \Delta T_v) \right) + \sum_{i \in I} (\rho_i (\Delta s_i + \Delta f_i)) \right)$$

La finalità è quella di minimizzare le spese complessive di servizio, compresi i costi fissi di utilizzo del veicolo, i costi legati al tempo e alla distanza di viaggio, i costi legati al tempo di servizio ed eventuali costi di attesa e penalità.

- Costo fisso per l'uso del mezzo: il parametro c_{f_v} rappresenta il costo fisso di utilizzo del veicolo v e la variabile binaria x_{pv} diventa uguale a uno solo se il mezzo v situato nel deposito p viene impiegato.
- Costo orario per l'uso del mezzo: il parametro TV_v indica la durata del tour, comprensiva dei tempi di viaggio, tempi di attesa e tempi di servizio, e viene moltiplicato per c_t , parametro indicante il costo di manodopera per unità di tempo.
- Costo della distanza: il parametro CV_v rappresenta il costo totale di viaggio, basato sulla distanza.
- Penalità violazione tempo massimo: il parametro $\rho_v \Delta T_v$ rappresenta la sanzione dovuta al superamento del tempo massimo di lavoro consentito.

- Penalità violazione finestre temporali: il parametro $\sum_{i \in I} \rho_i (\Delta s_i + \Delta f_i)$ rappresenta la sanzione dovuta alla violazione delle finestre temporali.

5.1.3. Vincoli del problema

La funzione obiettivo è soggetta ai seguenti vincoli:

1) Assegnazione dei nodi ai veicoli.

Ogni nodo $i \in I$ deve essere servito da un unico veicolo $v \in V$. E' vietato dividere il carico di un nodo su veicoli diversi.

$$\sum_{v \in V} Y_{iv} = 1 \quad \forall i \in I$$

2) Definizione dei veicoli nei depositi.

Ogni veicolo v impiegato deve essere situato in un unico deposito p al quale fa ritorno dopo aver concluso il tragitto assegnato. La dimensione della flotta richiesta è una variabile del problema da determinare simultaneamente con il miglior insieme di percorsi ed orari.

$$\sum_{p \in P} X_{pv} \leq 1 \quad \forall v \in V$$

3) Minor costo di viaggio per il veicolo v per arrivare al nodo i .

Il costo del viaggio dal deposito p al nodo i (C_i) deve essere maggiore o uguale a c_{pi}^v solo se il nodo $i \in I$ è servito dal veicolo v ($Y_{iv} = 1$) assegnato al deposito p ($X_{pv} = 1$). Questo perché, per definizione, c_{pi}^v indica il costo minimo di viaggio dal deposito p al nodo i . L'espressione seguente è vincolante solo nel caso in cui il cliente i è visitato per primo dal veicolo v .

$$C_i \geq c_{pi}^v (X_{pv} + Y_{iv} - 1) \quad \forall i \in I, p \in P, v \in V$$

4) Relazione tra i costi di viaggio dei nodi $i, j \in I$ sullo stesso percorso.

Posto che c_{ij}^v rappresenta il minor costo di viaggio dal nodo i al nodo j con il veicolo v , se entrambi i nodi (i, j) sono sullo stesso percorso ($Y_{iv} = Y_{jv} = 1$, per qualche veicolo v) ed il nodo i è visitato prima ($S_{ij} = 1$) allora l'espressione che segue stabilisce che il costo di viaggio basato sulla distanza dal deposito al nodo j (C_j) deve essere sempre maggiore di C_i di almeno c_{ij}^v .

$$a) \quad C_j \geq C_i + c_{ij}^v - M_c (1 - S_{ij}) - M_c (2 - Y_{iv} - Y_{jv}) \quad \forall i, j \in I, v \in V: i < j$$

Nel caso in cui il nodo j sia visitato prima del nodo i ($S_{ij} = 0$), vale l'istruzione inversa.

$$b) C_i \geq C_j + c_{ji}^v - M_c S_{ij} - M_c (2 - Y_{iv} - Y_{jv}) \forall i, j \in I, v \in V: i < j$$

Le due formulazioni appena esposte diventano vincoli ridondanti ogni volta che i nodi $i, j \in I$ sono serviti da veicoli diversi ($Y_{iv} + Y_{jv} < 2$, per ogni v). Per definizione M_c è un numero positivo elevato.

5) Costo di viaggio complessivo lungo il percorso assegnato al veicolo v .

Questo vincolo stabilisce che il costo complessivo di viaggio (CV_v) sostenuto dal mezzo v per completare il servizio di raccolta/consegna deve essere sempre maggiore delle spese di viaggio dal deposito ad ogni nodo i (C_i) lungo il percorso di almeno l'importo c_{ip}^v . L'ultimo nodo visitato dal veicolo v è quello che alla fine definisce il valore di CV_v . Pertanto il vincolo relativo a tale nodo e al veicolo assegnato v è proprio quello determinante all'ottimo.

$$CV_v \geq C_i + c_{ip}^v - M_c (2 - X_{pv} - Y_{iv}) \forall i \in I, p \in P, v \in V$$

6) Tempo iniziale di inizio servizio al nodo i .

Questo vincolo stabilisce che il veicolo v non può mai iniziare il servizio al nodo i assegnato prima del tempo t_{pi}^v (tempo minimo di viaggio dal deposito p al nodo i). Viene assunto che il veicolo v sia pronto in $t = 0$, altrimenti il tempo di partenza del veicolo v dovrebbe essere aggiunto a t_{pi}^v .

$$T_i \geq t_{pi}^v (X_{pv} + Y_{iv} - 1) \forall i \in I, p \in P, v \in V$$

7) Relazione tra i tempi di inizio servizio delle coppie di nodi (i, j) appartenenti allo stesso percorso.

Se si suppone che i nodi i e j siano entrambi serviti dallo stesso mezzo v , se il nodo i è visitato prima ($S_{ij} = 1$), la formulazione seguente afferma che il tempo di inizio servizio al nodo j (T_j) deve essere maggiore di T_i di almeno la somma del tempo di viaggio t_{ij}^v e del tempo di servizio al nodo i (st_i^v). Anche in questo caso M_t è un numero positivo elevato.

$$a) T_j \geq T_i + st_i^v + t_{ij}^v - M_t (1 - S_{ij}) - M_t (2 - Y_{iv} - Y_{jv}) \forall i, j \in I, v \in V: i < j$$

In caso contrario, se il nodo j è visitato prima ($S_{ij} = 0$), vale la formulazione inversa, esposta di seguito.

$$b) T_i \geq T_j + st_j^v + t_{ji}^v - M_t S_{ij} - M_t (2 - Y_{iv} - Y_{jv}) \forall i, j \in I, v \in V: i < j$$

Se uno dei due nodi non è sullo stesso percorso, allora $Y_{iv} + Y_{jv} < 2$ ed entrambe le formulazioni diventano ridondanti.

8) Tempo di viaggio complessivo per il veicolo v .

Il tempo totale richiesto dal veicolo v per completare il percorso viene trovato aggiungendo la somma del tempo di servizio al nodo i (st_i^v) ed il tempo di viaggio t_{ip}^v lungo l'arco (i, p) al tempo iniziale di servizio dell'ultimo nodo visitato i (cioè, il tempo di servizio iniziale più grande). Dal momento che l'ultimo nodo visitato dal veicolo v non è noto in anticipo, allora questo vincolo deve essere scritto per ogni nodo i .

$$TV_v \geq T_i + st_i^v + t_{ip}^v - M_t(2 - X_{pv} - Y_{iv}) \quad \forall i \in I, p \in P, v \in V$$

9) Violazione dei vincoli di tempo a causa del servizio in anticipo/in ritardo presso i clienti

Le finestre temporali possono essere rigide o flessibili. Quando le finestre temporali sono viste come vincoli rigidi, la formulazione seguente stabilisce che un veicolo v non può iniziare il servizio al nodo assegnato i prima del tempo iniziale s_i , ponendo semplicemente $\Delta s_i = 0$.

$$a) \quad \Delta s_i \geq s_i - T_i \quad \forall i \in I$$

Analogamente è vietato iniziare il servizio al nodo i dopo il tempo finale consentito f_i (impostando $\Delta f_i = 0$).

$$b) \quad \Delta f_i \geq T_i - f_i \quad \forall i \in I$$

Nel caso di finestre temporali flessibili, i vincoli precedentemente esposti possono essere violati dal mezzo v che può avviare il servizio al nodo i prima del tempo s_i , subendo un costo definito. In questo caso, le variabili Δs_i e Δf_i rappresentano la dimensione delle violazioni del vincolo della finestra temporale dovute rispettivamente al servizio anticipato o ritardato al nodo i .

10) Violazione del vincolo di tempo per il veicolo v .

Questo vincolo si applica solo nel caso in cui il tempo di lavoro massimo consentito tv_v^{max} è considerato come un vincolo flessibile che può essere violato ad un certo costo di penalità. Altrimenti, TV_v non dovrebbe essere maggiore di tv_v^{max} .

$$\Delta T_v \geq TV_v - tv_v^{max} \quad \forall v \in V$$

11) Vincoli di capacità

Il carico complessivo della raccolta/consegna di merce presso i clienti serviti da un veicolo v , non deve mai eccedere la capacità di carico q_v del mezzo stesso.

Ogni veicolo impiegato v è situato in un deposito p e, pertanto, $\sum_p X_{pv} = 1$.

$$\sum_{i \in I} w_i Y_{iv} \leq q_v \sum_{p \in P} X_{pv} \quad \forall v \in V$$

Le variabili continue del problema, T_i , TV_v , C_i , CV_v , Δs_i , Δf_i e ΔT_v , sono tutte non negative. I vincoli di assegnazione (punti 1, 2) insieme ai vincoli di costo del viaggio (punti 3, 5) ed i vincoli di capacità del mezzo (punto 11) definiscono tutto lo spazio delle soluzioni possibili per il VRP tradizionale. I vincoli di tempo includono i vincoli del tempo di visita (punti 6, 8) e le violazioni del vincolo di tempo flessibile (punti 9, 10). Non sono necessari vincoli di violazione dei sotto-percorsi.

I vincoli di lavoro e di finestra temporale possono essere considerati come vincoli rigidi semplicemente impostando a zero tutte le variabili (Δs_i , Δf_i e ΔT_v); operando così, i termini del costo di penalità vengono rimossi dalla funzione obiettivo. In breve, il modello matematico generale proposto può rappresentare una flotta di dimensione fissa o variabile composta da veicoli omogenei o eterogenei, situata in depositi singoli o multipli.

5.1.4. Regole di eliminazione esatta

Il CVRPTW può essere semplificato e ridotto di dimensione, usando l'informazione sulle finestre temporali dei clienti per migliorare l'efficienza dell'algoritmo di soluzione. Più le finestre temporali sono strette, maggiore è il numero delle variabili di sequenziamento e di vincoli che possono essere eliminati dalla formulazione del problema. Vengono descritte di seguito le regole di eliminazione legate ai concetti sopra esposti e che considerano le finestre temporali come vincoli rigidi.

Regola1

Se nessun veicolo $v \in V$ può servire una coppia di nodi $i, j \in I$ senza violare i vincoli delle relative finestre temporali, allora devono essere usati due diversi mezzi per servire i clienti i e j . Qualunque sia il nodo visitato per primo, il servizio all'altro nodo non può iniziare dopo il tempo di inizio finale a meno che venga usato un altro veicolo. Questo concetto viene evidenziato dalla seguente formulazione:

$$\forall i, j \in I, v \in V: i < j \wedge (s_i + st_i^v + t_{ij}^v) \geq f_j \wedge (s_j + st_j^v + t_{ji}^v) \geq f_i \Rightarrow Y_{jv} + Y_{iv} \leq 1$$

Quando per una coppia di nodi $i, j \in I$ vale la *Regola1*, allora il vincolo $Y_{jv} + Y_{iv} \leq 1$ deve essere incluso nella formulazione del problema perché il veicolo v può servire o il nodo i o il nodo j ma non entrambi. In questo caso i vincoli 4a), 4b), 7a) e 7b) possono essere eliminati per ogni coppia $(i, j) \in I$ in quanto soddisfano la *Regola1*. Inoltre, la variabile di sequenziamento S_{ij} non è più necessaria e perde di significato perché entrambi i nodi i, j non possono appartenere allo stesso percorso.

Regola2

Si suppone che i nodi $i, j \in I$ siano visitati dallo stesso veicolo v . Inoltre, la somma del tempo di inizio servizio al nodo i (s_i) ed il tempo di viaggio tra entrambi i nodi, incluso il tempo di servizio al nodo i , è maggiore del tempo di inizio servizio finale al nodo j (f_j). Allora il nodo i non può essere visitato prima del nodo j e vale che $S_{ij} = 0$, come espresso dalla seguente formulazione:

$$\forall i, j \in I, v \in V: i < j \wedge (s_i + st_i^v + t_{ij}^v) > f_j \wedge (s_j + st_j^v + t_{ji}^v) \leq f_i \Rightarrow S_{ij} = 0$$

Se vale la *Regola2*, allora il nodo i non può essere visitato prima del nodo j . Quindi la variabile di sequenziamento S_{ij} ed i vincoli 4a) e 7a) possono essere omessi dalla formulazione matematica del problema. Inoltre i vincoli 4b) e 7b) possono essere semplificati e riformulati rispettivamente nelle seguenti espressioni, applicate solo nel caso in cui entrambi i nodi appartengono allo stesso percorso, $Y_{iv} + Y_{jv} = 2$, altrimenti diventano ridondanti.

$$C_i \geq C_j + c_{ji}^v - M_c (2 - Y_{iv} - Y_{jv})$$

$$T_i \geq T_j + st_j^v + t_{ji}^v - M_t (2 - Y_{iv} - Y_{jv})$$

Regola3

Si suppone che i nodi $i, j \in I$ siano stati assegnati allo stesso percorso. Inoltre, la somma del tempo di inizio servizio iniziale al nodo j (s_j) e del tempo di viaggio lungo il percorso (i, j) , incluso il tempo di servizio al nodo j , è maggiore del tempo di inizio servizio finale al nodo i (f_i). Allora, il nodo j non può precedere il nodo i sullo stesso percorso e vale che $S_{ij} = 1$.

$$\forall i, j \in I, v \in V: i < j \wedge (s_i + st_i^v + t_{ij}^v) \leq f_j \wedge (s_j + st_j^v + t_{ji}^v) > f_i \Rightarrow S_{ij} = 1$$

In questo caso, i vincoli 4b) e 7b) possono essere eliminati dal modello e le formulazioni 4a) e 7a) si riducono rispettivamente a:

$$C_j \geq C_i + c_{ij}^v - M_c (2 - Y_{iv} - Y_{jv})$$

$$T_j \geq T_i + st_i^v + t_{ij}^v - M_t(2 - Y_{iv} - Y_{jv})$$

5.2. Algoritmo euristico cluster

Il *CVRPTW* con una flotta eterogenea e depositi multipli è molto complicato da risolvere attraverso un puro approccio di ottimizzazione. Infatti, perfino i *Vehicle Routing Problems* più semplici sono tra le classi più difficili di problemi di ottimizzazione combinatoria. Di seguito, viene introdotto un algoritmo euristico basato sulle finestre temporali che assembla efficientemente i nodi dei clienti in un numero piuttosto basso di cluster fattibili. Esprimendo il modello matematico in termini di pochi cluster, anziché un numero enorme di clienti, la dimensione del *CVRPTW* può essere bruscamente ridotta. Inoltre, viene abbassato in modo massiccio l'onere computazionale delle fasi di risoluzione successiva per trovare i *routing*, in quanto, l'euristica di *Cheapest Insertion* verrà poi applicata ai singoli cluster (pochi nodi) in modo distinto e non univocamente all'intero parco clienti.

5.2.1. Procedura definizione cluster

Per trovare rapidamente un buon insieme di cluster per un *CVRPTW* di grandi dimensioni, viene applicata la procedura euristica di seguito riportata.

Passo_1

- a. Si apre una lista di nodi L e li si ordina per valori crescenti dei tempi di inizio servizio s_i (limite inferiore della finestra temporale $[s_i, f_i]$). Se alcuni nodi hanno lo stesso s_i , li si organizza per valori crescenti del tempo f_i (limite superiore della finestra temporale $[s_i, f_i]$).
- b. Si apre una lista di veicoli disponibili V e li si ordina per valori decrescenti del rapporto (q_v / cf_v) .
- c. Si sceglie la distanza massima consentita tra ogni coppia di nodi nello stesso cluster (d^{max}) ed il tempo di attesa massimo consentito (Δ).

Passo_2

Si apre una lista vuota K_n legata al prossimo cluster C_n da creare. Si assegna la prima voce della lista V al cluster C_n e la si elimina da V .

Passo_3

- a. Si prende il primo nodo i dalla lista L e lo si posiziona in fondo alla lista K_n . Si valorizzano i parametri del cluster C_n , come di seguito esposto (il simbolo " \leftarrow "

indica che il valore dell'espressione a destra dello stesso simbolo viene copiato all'interno della variabile a sinistra dello stesso):

aC_n : tempo di inizio servizio iniziale del cluster C_n

$$aC_n \leftarrow s_i$$

bC_n : tempo di inizio servizio finale del cluster C_n

$$bC_n \leftarrow f_i$$

wC_n : carico corrente da raccogliere presso il cluster C_n

$$wC_n \leftarrow w_i$$

stC_n : tempo di servizio corrente presso il cluster C_n

$$stC_n \leftarrow st_i$$

- b. Si elimina il nodo i dalla lista L e si fa una copia della lista corrente L la si chiama L' .

Passo_4

Si prende il primo nodo j dalla lista L' e si verifica che il carico corrente da raccogliere dal cluster C_n sommato a w_j non ecceda la capacità di carico q_v del veicolo assegnato v . Se la capacità del veicolo viene superata si elimina il nodo j dalla lista L' e si ripete il *Passo_4*, altrimenti, si procede con il passo successivo.

Passo_5

- a. Si calcola la distanza d_{ji} tra il nodo j ed il suo nodo più vicino i sulla lista K_n .
- b. Prima di passare al punto successivo, si verifica che d_{ji} sia più piccola della distanza massima consentita d^{max} , altrimenti si elimina il nodo j dalla lista L' e si torna al *Passo_4*.

Passo_6

Per proseguire la procedura bisogna superare positivamente il seguente vincolo $aC_n + stC_n + t_{ij}^v \leq \max(bC_n, f_j)$ altrimenti si elimina il nodo j dalla lista L' e si torna al *Passo_4*.

Passo_7

Per proseguire la procedura bisogna superare positivamente il seguente vincolo $aC_n + stC_n + t_{ij}^v + \Delta \geq s_j$ altrimenti si chiude il cluster C_n , si elimina la lista temporanea L' , si salva la lista K_n , si definisce C_n e si torna al *Passo_2*.

Passo_8

- a. Si posiziona il nodo j in fondo alla lista K_n e si aggiornano i parametri del cluster C_n come segue:

$$wC_n \leftarrow wC_n + w_j \quad stC_n \leftarrow \max(stC_n + t'_{ij} + st_j, s_j + st_j - s_i)$$

- b. Se $bC_n > f_j$ allora $bC_n \leftarrow f_j$ altrimenti il tempo di arrivo finale bC_n rimane invariato. Si elimina il nodo j dalla lista L e L' e si prosegue la procedura.

Passo_9

- a. Se la lista L' è vuota, prima di procedere al passo successivo, si salva la lista K_n definendo il cluster C_n , altrimenti si torna al *Passo_4*.

Passo_10

Si ripetono i passaggi dal *Passo_2* al *Passo_9* fino a che la lista L non diventa vuota.

5.2.1. Spiegazione dettagliata dell'algoritmo

Gli input della procedura sono l'insieme dei nodi i, j dei clienti I , l'insieme dei veicoli V , le distanze (o i costi) ed i tempi di viaggio tra i nodi, i tempi di servizio ed i carichi raccolti/consegnati oltre ai dati sulle finestre temporali. Lo scopo della procedura è quello di accorpate tutti i clienti in un numero limitato di cluster fattibili che rispettino quanto segue:

- il carico del cluster è assegnato ad un singolo veicolo;
- esiste un percorso che connette i nodi del cluster e soddisfa tutti i vincoli delle finestre temporali dei clienti interessati dal raggruppamento;
- il tempo di attesa del veicolo a causa dell'arrivo anticipato ai punti di raccolta/consegna deve essere mantenuto il più piccolo possibile;
- la lunghezza media per nodo percorsa dal veicolo assegnato per tutto il cluster deve rimanere bassa.

Per raggiungere questi obiettivi, la lista dei nodi L viene opportunamente disposta nel *Passo_1* per facilitare la generazione di cluster convenienti e fattibili. Prima di aggiungere un altro nodo al cluster generato, vengono testate la sua vicinanza agli altri nodi del cluster (*Passo_5*) e la soddisfazione dei vincoli di capacità del veicolo (*Passo_4*) e di finestra temporale (*Passo_6* e *Passo_7*). Inoltre, è consentito solo un tempo di ozio massimo (Δ) per l'arrivo anticipato al sito del cliente j (cioè prima del tempo di inizio servizio iniziale s_j) altrimenti l'incorporazione del nodo j nel cluster viene respinta (*Passo_7*).

Per definire la formulazione matematica in termini di cluster, anziché di nodi, sono state introdotte le due nozioni seguenti:

➤ **Finestra temporale del cluster**

Il tempo di inizio servizio iniziale (aC_n) al cluster C_n è dato dal valore minimo di s_i , con $i \in C_n$, mentre il tempo di inizio servizio finale (bC_n) al cluster C_n viene aggiornato, di volta in volta, prendendo il minimo tra il valore di bC_n (tempo di inizio servizio finale al cluster C_n corrente) ed il valore di f_j (tempo di inizio servizio finale al cliente j , nuovo entrato nel cluster C_n). In questo modo, le finestre temporali per i nodi correnti in C_n sono tutte soddisfatte e le finestre temporali del cluster diventano, dopo ogni inserimento, più strette (*Passo_8.b*).

➤ **Tempo di servizio del cluster**

Viene indicato con stC_n e rappresenta una buona approssimazione del tempo complessivo speso dal veicolo assegnato durante la visita al cluster C_n (include non solo il tempo di servizio ma anche i tempi di ozio e di viaggio in tutti i nodi contenuti nel cluster).

Dato che la dimensione della flotta è una variabile del problema, la procedura deve scegliere i veicoli più efficienti (*Passo_2*) per assolvere i compiti di raccolta/consegna; per ottenere questo, la lista dei mezzi viene ordinata secondo criteri crescenti di economicità (*Passo_1.b*).

I dati dei clienti contenuti nei cluster, ottenuti mediante questa procedura, diventano elementi di input distinti per la fase successiva di inserzione; l'algoritmo euristico di *Cheapest Insertion*, descritto nel capitolo successivo, mira a definire il sequenziamento (*route*) dei clienti contenuti all'interno di ciascun cluster, a cui è già stato assegnato il veicolo.

CAPITOLO 6

L'algoritmo euristico Cheapest Insertion per la soluzione del CVRPTW

Dopo aver visto la prima parte del metodo euristico a due fasi di tipo *Cluster first – Route second*, viene ora esplicitata la modalità di costruzione dei *route* ammissibili associati a ciascun cluster precedentemente individuato.

In questo capitolo vengono presentate le basi teoriche della procedura euristica costruttiva – sequenziale di *Cheapest Insertion* che mira a definire il sequenziamento (*route*) dei clienti contenuti all'interno di ciascun cluster. Al raggruppamento di clienti è già stato assegnato un veicolo (nella fase precedente di clustering) il cui tragitto viene ora costruito mediante l'iterazione dell'inserimento del cliente che presenta il costo di inserzione minore. I punti fondamentali che riguardano lo sviluppo di questo algoritmo euristico sono di seguito esposti.

6.1. Inizializzazione

E' la prima fase dell'algoritmo e prevede l'inserimento nel tragitto del primo nodo (macello) dopo il deposito: questo cliente deve essere quello che genera il minor costo nel tragitto tra il deposito ed il cliente stesso.

Partendo dai dati di input, si calcolano le voci di costo (spiegate di seguito in questo paragrafo) per ogni coppia di macelli e per ogni camion, dal momento che per una stessa coppia di nodi il costo varia a seconda del mezzo impiegato. Il costo minore viene individuato tra tutte le coppie Deposito – Macello, mentre i valori calcolati per tutte le altre coppie Macello – Macello rimangono utili per i successivi inserimenti, a patto che tali valori siano aggiornati correttamente al momento del loro utilizzo. Per la descrizione delle voci di costo, coinvolte in questa prima fase, si considerano due clienti generici i, j ed un generico mezzo k tra gli r a disposizione.

➤ Tempo Totale

Questo elemento risulta necessario per il calcolo della prima voce di costo. E' formato dal tempo di viaggio tra i due nodi ed il tempo di carico del nodo di partenza, riferiti al mezzo in considerazione.

$$\text{Tempo Totale}_k(i, j) = \text{Tempo di Viaggio}_k(i, j) + \text{Tempo di Carico}_k(i)$$

➤ **Costo di Attesa**

Sorge solamente quando il veicolo, recandosi dal cliente i al cliente j , arriva in j prima dell'orario di apertura (s_j) ed è costretto ad aspettare, spreca così del tempo impiegabile per il servizio ad altri nodi. Questo tempo sprecato genera logicamente un costo in termini di costo unitario di lavoro; quando invece il veicolo giunge al macello in corrispondenza (o successivamente) alla sua apertura, il costo d'attesa è nullo.

Se

$$\text{Tempo Totale}_k(i, j) < \text{Apertura}(j)$$

allora

$$\text{Tempo di Attesa}_k(i, j) = (\text{Apertura}(j) - \text{Tempo Totale}_k(i, j))$$

$$\text{Costo di Attesa}_k(i, j) = \text{Tempo di Attesa}_k(i, j) \times \text{Costo Unitario Tempo}_k$$

Altrimenti

$$\text{Tempo di Attesa}_k(i, j) = 0$$

$$\text{Costo di Attesa}_k(i, j) = 0$$

➤ **Costo di Viaggio**

Indica quanto costa lo spostamento tra un macello ed il successivo. E' costituito da una componente "fissa" e da una componente "variabile" che esprime il costo del tempo impiegato per il transito.

$$\text{Costo di Viaggio}_k(i, j) =$$

$$\text{Costo Fisso di Viaggio}_k(i, j) + \text{Tempo di Viaggio}_k(i, j) \times \text{Costo Unitario Tempo}_k$$

➤ **Costo di Carico**

Esprime il costo necessario per effettuare il carico o lo scarico del veicolo presso il macello considerato; è costituito da una componente "fissa" e da una componente "variabile" che esprime il costo del tempo impiegato per il carico o lo scarico. Tale costo è fisso per uno specifico cliente, ma varia a seconda del mezzo impiegato.

$$\text{Costo di Carico}_k(i) =$$

$$\text{Costo Fisso di Carico}_k(i) + \text{Tempo di Carico}_k(i) \times \text{Costo Unitario Tempo}_k$$

➤ **Costo Totale**

E' la voce di costo analizzata dall'algorithm nella fase di inizializzazione per stabilire il macello di partenza da inserire subito dopo il deposito e nella fase successiva di assegnazione per valutare i costi di inserzione. E' dato dalla somma di tutte le voci di costo precedentemente calcolate aggiunta al costo fisso d'uscita del mezzo.

$$\begin{aligned} \text{Costo Totale}_k(i, j) = & \\ & \text{Costo Fisso Uscita}_k + \text{Costo di Carico}_k(i) + \\ & \text{Costo di Viaggio}_k(i, j) + \text{Costo di Attesa}_k(i, j) \end{aligned}$$

Dopo il calcolo di tutti i costi totali come descritto, si individua quale macello comporta il minor costo dal deposito e teoricamente per quale camion si genera tale costo. (Quest'ultima operazione non viene eseguita nel caso studio di questo elaborato in quanto il veicolo risulta essere già assegnato al cluster, e quindi ai clienti). Dopo aver inserito tra il deposito e la conceria il primo macello, individuato in questa fase di inizializzazione, l'algorithm procede con la successiva fase di assegnazione.

6.2. Assegnazione

Dopo la fase di inizializzazione, il tragitto per il primo camion selezionato si presenta come espresso in figura 6.1.



Figura 6.1 Tragitto veicolo dopo fase di inizializzazione

L'algorithm di inserzione verifica se è possibile inserire un macello, non ancora assegnato al tragitto, fra una delle coppie di nodi adiacenti individuabili nel percorso: in questo caso, essendo alla seconda iterazione, la procedura prevede la valutazione di un possibile inserimento di un cliente, non ancora assegnato, tra la coppia Deposito – Primo Macello e la coppia Primo Macello – Conceria. La discriminante che determina il nuovo nodo da inserire è rappresentata dalla minor voce di costo di inserzione, calcolata come somma tra il costo totale che si genererebbe per transitare dal primo nodo della coppia considerata al nodo candidato per l'inserzione ed il costo totale che si creerebbe per andare da quest'ultimo al secondo nodo della coppia, a cui viene poi tolto il costo totale riguardante il tragitto tra i due vertici adiacenti presi in esame. Ad

ogni nuovo inserimento aumenta il numero delle coppie di nodi adiacenti da esaminare ed il calcolo appena riportato va eseguito per ognuna di esse.

Data una coppia di nodi adiacenti (i, j) appartenente ad un determinato percorso ed un nodo u non ancora inserito, il costo di inserzione di u si calcola come segue:

$$\text{Costo di inserzione } (u) = \text{Costo Totale } (i, u) + \text{Costo Totale } (u, j) - \text{Costo Totale } (i, j)$$

Un volta trovato il costo di inserzione minore relativo al macello u , si conoscono all'interno del percorso i macelli adiacenti i e j attivi nella generazione di tale costo e l'algoritmo procede con l'inserimento nel tragitto in fase di costruzione del nuovo cliente u tra la coppia (i, j) individuata.

Dopo l'assegnazione, per ogni macello presente nel percorso vengono calcolate le variabili riportate di seguito, utili alla verifica dei vincoli (discussa nel paragrafo successivo) ed al calcolo del costo finale del problema.

✓ **Capacità Residua**

Dopo ogni iterazione (nuovo inserimento) la capacità massima inizialmente attribuita al veicolo viene ridotta di una quantità pari a quella richiesta dal singolo cliente inserito. La *Capacità Residua* indica, dopo l'inserimento, quanta capacità può soddisfare ancora il mezzo; dato un macello i appartenente al percorso vale l'espressione seguente:

$$\text{Capacità Residua } (i) = \text{Capacità Residua } (i - 1) - \text{Richiesta } (i)$$

La capacità residua del primo nodo del percorso (Deposito del mezzo) viene inizializzata con il valore della capacità totale del veicolo k in considerazione.

✓ **Tempo di Carico Assegnato**

Viene calcolato come somma del tempo di carico del macello considerato e di tutti i tempi di carico dei macelli precedenti, appartenenti al percorso associato al veicolo in considerazione.

$$\begin{aligned} \text{Tempo di Carico Assegnato } (i) = \\ \text{Tempo di Carico Assegnato } (i - 1) + \text{Tempo di Carico}_k (i) \end{aligned}$$

✓ **Tempo di Viaggio Assegnato**

Viene calcolato come somma del tempo di viaggio tra il macello considerato ed il precedente e di tutti i tempi di viaggio tra le coppie di macelli ad esso precedenti, appartenenti al percorso associato al veicolo in considerazione.

$$\begin{aligned} \text{Tempo di Viaggio Assegnato } (i) &= \\ \text{Tempo di Viaggio Assegnato } (i - 1) &+ \text{Tempo di Viaggio}_k (i-1, i) \end{aligned}$$

✓ **Tempo di Attesa Assegnato**

Viene calcolato come somma dei tempi di attesa di tutte le coppie presenti nel tragitto fino al macello i selezionato. Per il calcolo del tempo di attesa bisogna fare riferimento alla formulazione espressa nel paragrafo precedente.

$$\begin{aligned} \text{Tempo di Attesa Assegnato } (i) &= \\ \text{Tempo di Attesa Assegnato } (i - 1) &+ \text{Tempo di Attesa } (i-1, i) \end{aligned}$$

✓ **Tempo di Fine Servizio**

Indica il momento in cui il veicolo termina il servizio assegnato presso il cliente in questione; è ottenuto dalla somma del *Tempo di Carico Assegnato*, del *Tempo di Viaggio Assegnato* e dell'eventuale *Tempo di Attesa Assegnato*, tutte voci precedentemente calcolate relative al macello i in esame.

$$\begin{aligned} \text{Tempo di Fine Servizio } (i) &= \\ \text{Tempo di Carico Assegnato } (i) &+ \\ \text{Tempo di Viaggio Assegnato } (i) &+ \\ \text{Tempo di Attesa Assegnato } (i) & \end{aligned}$$

✓ **Costo di Carico Assegnato**

Viene calcolato come somma del costo di carico del macello considerato e di tutti i costi di carico dei macelli precedenti, appartenenti al percorso associato al veicolo in considerazione.

$$\begin{aligned} \text{Costo di Carico Assegnato } (i) &= \\ \text{Costo di Carico Assegnato } (i - 1) &+ \text{Costo di Carico}_k (i) \end{aligned}$$

✓ **Costo di Viaggio Assegnato**

Viene calcolato come somma del costo di viaggio tra il macello considerato ed il precedente e di tutti i costi di viaggio tra le coppie di macelli ad esso precedenti, appartenenti al percorso associato al veicolo in considerazione.

Costo di Viaggio Assegnato (i) =

Costo di Viaggio Assegnato (i - 1) + Costo di Viaggio_k (i-1, i)

Dopo ogni inserimento è fondamentale aggiornare il costo totale per tutte le coppie di macelli presenti nel problema (assegnati e non assegnati), in particolare va focalizzata l'attenzione sulla componente del costo d'attesa, unica voce del costo totale che può aver subito variazioni. Dal momento che l'inserimento di un macello nel percorso va ad incrementare i tempi di fine servizio di altri macelli presenti, può verificarsi la riduzione (o addirittura l'annullamento) della differenza di tempo che c'era in precedenza (quindi prima dell'inserimento) tra la fine servizio in un macello ed il momento di apertura di un altro, andando a modificare conseguentemente il tempo di attesa ed il relativo costo. Le altre componenti del costo totale, inerenti all'uscita del veicolo, al carico ed al viaggio, un volta calcolate, rimangono sempre inalterate poiché non possono variare la distanza ed il tempo di viaggio tra due nodi.

Le operazioni appena descritte completano la fase di assegnazione che, per avere esito positivo e fattibile, necessita della verifica dei vincoli (descritta nel paragrafo successivo). Infatti, non è detto che ogni assegnazione risulti accettabile: il nuovo macello inserito può far sì che qualche cliente, appartenente al tragitto, non sia più in grado di superare i vincoli imposti dal problema. A tal proposito, dopo ogni inserimento, l'algoritmo prende in esame tutti i macelli del percorso e verifica che ognuno di essi rispetti i vincoli. Nel caso in cui un nodo presenti condizioni non accettabili, come il non rispetto di un solo vincolo, la procedura va a rimuovere dal percorso il cliente precedentemente assegnato e candida al nuovo possibile inserimento il macello che presenta il secondo minor costo di inserzione. Se questa nuova assegnazione dovesse avere esito negativo, il nuovo candidato all'inserimento diventerebbe quello che presenta il terzo minor costo di inserzione e così via fino a quando l'assegnazione diventa compatibile con i vincoli o terminano i candidati all'inserimento. Nell'ultimo caso, il tragitto in costruzione per il veicolo considerato viene chiuso e la procedura riparte con l'inizializzazione di un nuovo percorso abbinato ad un altro mezzo del deposito.

6.3.Verifica dei vincoli

Dopo l'inserimento di un nuovo macello al tragitto, avviene la verifica dei vincoli appartenenti ai tre gruppi già introdotti nel paragrafo 4.3.2; di seguito sono descritte le tre condizioni restrittive per decretare o meno l'esito positivo dell'assegnazione appena avvenuta.

- **Vincolo di capacità**

Dopo l'assegnazione, la *Capacità Residua* di ogni macello appartenente al tragitto deve rimanere positiva; nel caso avverso significa che il veicolo abbinato al tragitto non possiede una capacità tale da poter soddisfare la totalità della domanda dei clienti appartenenti al tragitto stesso.

Per ogni cliente del percorso deve valere che:

$$Capacità\ Residua > 0$$

- **Vincolo di tempo di chiusura del macello**

Dopo l'assegnazione, il *Tempo di Fine Servizio* di ogni macello deve essere inferiore o al limite uguale al suo tempo di chiusura, altrimenti significa che il veicolo non è in grado di completare le operazioni di raccolta/consegna nei tempi stabiliti e l'ultimo macello inserito deve essere rimosso.

Per ogni cliente del percorso deve valere che:

$$Tempo\ di\ Fine\ Servizio < Chiusura$$

- **Vincolo di tempo massimo di scarico**

Un cliente può avere la necessità di far giungere le proprie merci deperibili in un determinato posto, in questo caso in conceria, entro un tempo massimo. Quest'ultimo deve essere sempre superiore al tempo di fine servizio del veicolo presso il cliente stesso che ha manifestato tale necessità, altrimenti bisogna rimuovere l'ultimo nodo inserito per non incorrere nel superamento del *Tempo Massimo di Scarico*.

Per ogni cliente del percorso deve valere che:

$$Tempo\ di\ Fine\ Servizio < Tempo\ Massimo\ di\ Scarico$$

Questi sono i vincoli che tutti i clienti del percorso devono rispettare dopo ogni inserzione, pena la rimozione dell'ultimo nodo inserito e la sostituzione di questo con il candidato che presenta il successivo minor costo di inserzione.

6.4. Soluzione del CVRPTW e calcolo del costo finale totale

Un volta che tutti i clienti sono stati inseriti, o è stato assegnato un tragitto a tutti i mezzi disponibili, l'algoritmo genera una tabella riepilogativa per ogni veicolo, riportante tutti i nodi assegnati al suo percorso e per ciascuno di questi esplicita i valori delle variabili descritte nel paragrafo 6.2, utili al calcolo del costo finale di ogni tragitto. La soluzione finale del *CVRPTW*, ossia il costo finale totale del problema, è costituita dalla somma dei costi finali di ogni singolo tragitto individuato dalla procedura.

Il costo finale di ogni route è dato dalla somma dei seguenti componenti:

- Costo di uscita del camion che parte dal deposito e percorre quel tragitto;
- Costo fisso di viaggio totale: è ottenuto mediante la somma dei costi di viaggio per ogni tratta del percorso e, per quanto descritto prima, coincide con la variabile *Costo di Viaggio Assegnato* relativa all'ultimo nodo del percorso (conceria).
- Costo di carico totale: è ottenuto mediante la somma dei costi di carico sostenuti presso ogni cliente del percorso e, per quanto descritto prima, coincide con la variabile *Costo di Carico Assegnato* relativa all'ultimo nodo del percorso (conceria).
- Costo di attesa per il primo macello visitato: è riferito all'eventuale tempo di attesa che si può generare quando il veicolo giunge nel primo nodo visitato del percorso con anticipo rispetto all'orario di apertura dello stesso. In questo caso il mezzo ritarda la partenza dal deposito fino a quando il suo orario di arrivo previsto presso il primo nodo visitato non coincide con l'orario di apertura del medesimo. Ogni veicolo ha però un tempo massimo entro il quale partire, di conseguenza non può attendere oltre tale tempo. Nel caso in cui il tempo di attesa sia così elevato da superare il tempo massimo di partenza, viene tenuto fermo il più possibile il veicolo presso il deposito, per poi farlo attendere per il tempo rimanente presso il primo nodo visitato.

Se

$\text{Tempo Max Partenza}_k >$

$\text{Tempo Min Partenza}_k + \text{Tempo di Attesa}_k (\text{Dep, Primo Mac})$

Allora

$\text{Costo Attesa Primo Mac} =$

$\text{Tempo di Attesa}_k (\text{Dep, Primo Mac}) \times \text{Costo Start Delay}_k$

Altrimenti

$\text{Costo Attesa Primo Mac} =$

$(\text{Tempo Max Partenza}_k - \text{Tempo Min Partenza}_k) \times \text{Costo Start Delay}_k +$

$(\text{Apertura (Primo Mac)} - (\text{Tempo Max Partenza}_k +$

$\text{Tempo di Viaggio}_k (\text{Dep, Primo Mac})) \times \text{Costo Unitario Tempo}_k$

Nel primo caso il mezzo, per convenienza, trascorre il tempo all'interno del deposito per generare un costo d'attesa minore; nel secondo, invece, il veicolo attende presso il deposito per il maggior tempo possibile (fino al suo tempo di partenza massimo) e poi termina l'attesa presso il primo nodo visitato.

- Costo del tempo di viaggio totale: è il costo generato dal tempo totale impiegato dal mezzo per gli spostamenti da un cliente all'altro. Viene calcolato con due modalità, a seconda che ci sia o meno attesa nel deposito da parte del veicolo prima di raggiungere il primo nodo visitato.

❖ Caso 1: $\text{Tempo di Attesa}_k (\text{Dep, Primo Mac}) > 0$

Se c'è attesa, il costo del tempo di viaggio totale è dato dal prodotto tra il *Costo Unitario Tempo_k* e la differenza tra il *Tempo di Fine Servizio* dell'ultimo nodo del tragitto (conceria) ed il tempo di attesa tra il deposito ed il primo nodo servito.

$\text{Costo Tempo Viaggio Totale} =$

$\text{Costo Unitario Tempo}_k \times$

$(\text{Tempo di Fine Servizio (conceria)} -$

$\text{Tempo di Attesa}_k (\text{Dep, Primo Mac}))$

❖ Caso 2: $Tempo\ di\ Attesa_k (Dep, Primo\ Mac)=0$

Se non c'è attesa, il costo del tempo di viaggio totale è dato dal prodotto tra il *Costo Unitario Tempo_k* e la differenza tra il *Tempo di Fine Servizio* dell'ultimo nodo del tragitto (conceria) ed il *Tempo di Carico* del deposito. Quest'ultimo valore è pari al tempo di partenza minimo dal deposito del veicolo *k* (*Tempo Min Partenza_k*)

$$\begin{aligned} \text{Costo Tempo Viaggio Totale} = \\ \text{Costo Unitario Tempo}_k \times \\ (\text{Tempo di Fine Servizio (conceria)} - \\ \text{Tempo Min Partenza}_k) \end{aligned}$$

La somma di queste cinque componenti porta al valore del *Costo Finale Totale* di un tragitto, come evidenziato dalla formulazione seguente:

$$\begin{aligned} \text{Costo Finale Totale} = \\ \text{Costo Uscita}_k + \\ \text{Costo Fisso Viaggio Totale} + \\ \text{Costo Carico Totale} + \\ \text{Costo Attesa Primo Mac} + \\ \text{Costo Tempo Viaggio Totale} \end{aligned}$$

La somma di tutte le variabili *Costo Finale Totale*, calcolate per ogni tragitto individuato dall'algoritmo, fornisce la soluzione del *CVRPTW* e quindi il costo finale del problema.

CAPITOLO 7

Implementazione in Excel – VBA dell’algoritmo Cluster con Cheapest Insertion

Questo elaborato riguarda il *CVRPTW* legato al trasporto di pelli bovine fresche da dei macelli (dove sono state lavorate) ad una conceria. L’obiettivo è quello di trovare una soluzione buona, e quindi, differentemente dagli algoritmi esatti, non una soluzione ottima, ma una che si avvicini a questa e sia ottenuta con brevi tempi di risposta impiegando applicativi di larga diffusione.

Dopo la spiegazione dettagliata delle basi teoriche degli algoritmi euristici *Cluster* e *Cheapest Insertion*, viene di seguito descritta la loro implementazione, mediante linguaggio di programmazione *Visual Basic for Applications (VBA)* in ambiente *Microsoft Office* (pacchetto applicativo *Excel*), per la soluzione del caso oggetto di questa tesi, già anticipato nel capitolo 4.

L’intero progetto è diviso in quattro blocchi, che, rispettivamente, rispondono ad esigenze specifiche, quali:

- inserimento dati di input del problema mediante interfaccia utente;
- raggruppamento di tutti i clienti in cluster ammissibili per ridurre la dimensione del problema;
- sequenziamento ottimale, mediante *Cheapest Insertion*, dei nodi contenuti nei cluster precedentemente individuati;
- definizione della soluzione finale del problema.

Di seguito, sono spiegati i passaggi appena esposti e la loro applicazione ad un contesto pratico composto da uno o più depositi, da cui possono partire 3 veicoli per soddisfare i fabbisogni e le richieste di 13 macelli; resta inteso che l’intero algoritmo scritto in *VBA* è stato sviluppato in modo parametrico, quindi adattabile a situazioni differenti tra loro. Alla fine di questo capitolo, infatti, il campo d’applicazione del progetto viene esteso ad un contesto costituito da 20 macelli e 5 veicoli.

7.1. Inserimento dati del problema relativo a 13 macelli e 3 camion

L'applicativo scritto in VBA è depositato all'interno di un modulo di un file *Excel*, costituito da un unico foglio di lavoro nominato "INIZIO" e contenente la pagina di presentazione, come illustrato in figura 7.1



Università degli studi di Padova
Corso di Laurea Magistrale in Ingegneria Gestionale

Tecniche Euristiche di Clusterizzazione e Percorso Ottimo per risolvere un problema di raccolta e consegna di materiale

RELATORE: CH.MO PROF. GIORGIO ROMANIN JACUR
LAUREANDO: ADRIANO CORRENTE

Anno Accademico 2013/2014

INIZIO

Figura 7.1 Schermata iniziale dell'algoritmo

Selezionando il tasto "INIZIO" si accede all'interfaccia utente, riportata di seguito in figura 7.2, con la quale è possibile inserire il numero dei clienti che devono essere soddisfatti ed il numero dei veicoli a disposizione per il trasporto del materiale.



Inserimento Dati

CLUSTER con CHEAPEST INSERTION

Tecniche euristiche per un problema di routing con finestre temporali, flotta eterogenea a capacità limitata

Inserisci il numero di MACELLI

Inserisci il numero di CAMION

CONFERMA

Figura 7.2 Interfaccia utente

Come anticipato, viene analizzato in questo caso studio un contesto pratico formato da 13 macelli e 3 veicoli; basta inserire questi due valori nelle rispettive *TextBox* e selezionare il pulsanti di comando “CONFERMA” per parametrizzare e dimensionare correttamente tutte le successive tabelle, necessarie per la memorizzazione dei dati di input del problema. I campi delle due *TextBox* devono essere chiaramente alimentati da valori in formato numerico e tale controllo viene fatto impiegando la funzione *IsNumeric()* standard in *VBA*, come riportato in Appendice, sezione A.1.2.

L'evento collegato alla selezione del pulsante di comando “CONFERMA” comporta il salvataggio dei valori delle *TextBox* all'interno delle variabili *n_macelli* e *n_camion*, la chiusura della *Form* interfaccia utente, la dichiarazione di tutte le variabili usate all'interno dell'applicativo (come indicato in Appendice, sezione A.1.1) e l'aggiunta di un nuovo foglio di lavoro al *Workbook* nominato “Dati_Input”, destinato a contenere tutte le tabelle dei dati del problema (come indicato in Appendice, sezione A.1.3). All'utente viene poi chiesto l'inserimento e la conferma dei dati del problema, in modo sequenziale ma diviso in sei sezioni distinte, come descritto nei paragrafi seguenti; il relativo codice *VBA* è riportato in Appendice, sezione A.1.4. Una precisazione, i valori posti a zero ed evidenziati in rosso all'interno delle tabelle di questo elaborato, sono privi di significato e per tale motivo non considerati durante le fasi successive di calcolo.

7.1.1. Dati riguardanti i Macelli e la Conceria

I primi dati da inserire sono quelli riportati in tabella 7.1, relativi ai macelli ed alla conceria. Se la prima riga è riservata al deposito e l'ultima alla conceria, in ciascuna delle righe rimanenti sono indicate le informazioni di uno specifico macello; le quattro colonne, invece, identificano i seguenti elementi:

- **Richiesta [Kg]:** è la quantità espressa in chilogrammi di pelli che deve essere trasportata dal macello alla conceria mediante un unico veicolo (equivale alla variabile p_i descritta nel paragrafo 4.2).
- **Tempo Max Scarico [h]:** indica il numero di ore entro il quale le pelli devono essere portate in conceria per essere lavorate senza incorrere in deperimento (equivale alla variabile g_i descritta nel paragrafo 4.2).
- **Apertura [h]:** è il limite inferiore della finestra temporale e definisce l'istante a partire dal quale il veicolo può iniziare il servizio (equivale alla variabile s_i descritta nel paragrafo 4.2).

- **Chiusura [h]:** è il limite superiore della finestra temporale e definisce l'istante entro il quale il veicolo deve terminare il servizio (equivale alla variabile f_i descritta nel paragrafo 4.2).

	Richiesta [Kg]	Tempo Max Scarico [h]	Apertura [h]	Chiusura [h]
Deposito	0	0	0	0
Macello_1	250	1700	120	540
Macello_2	530	1800	380	960
Macello_3	800	1800	180	490
Macello_4	300	1900	450	840
Macello_5	390	1701	210	900
Macello_6	340	1600	500	1500
Macello_7	420	1900	880	1400
Macello_8	390	1500	590	1000
Macello_9	480	1600	110	710
Macello_10	620	1800	720	1060
Macello_11	400	1700	100	1000
Macello_12	550	1700	260	1300
Macello_13	270	1500	80	1440
Conceria	0	0	420	1740

Tabella 7.1 Inserimento dati dei macelli e della conceria

L'utente dopo aver inserito le informazioni, deve confermarle mediante l'apposito pulsante di comando "Clicca per CONFERMA DATI" che attiva l'evento associato alla routine di controllo, come riportato in Appendice, sezione A.1.4. Solamente nel caso di esito positivo della verifica, le informazioni inerenti i macelli e la conceria vengono salvate in un array chiamato *ListaMacelli()* di dimensioni $n_macelli+2$ e di tipo *macello* come indicato nell'Appendice relativa alle dichiarazioni delle variabili, sezione A.1.1.

7.1.2. Dati riguardanti i Camion

In tabella 7.2 sono riportati i dati di input relativi ai mezzi presenti nel deposito e disponibili per il trasporto delle merci. In ogni riga sono riportate le informazioni di uno specifico veicolo mentre le sei colonne identificano i seguenti elementi:

- **Costo Unitario Tempo [€/h]:** è specifico per ogni mezzo ed indica la quantità di denaro nell'unità di tempo da considerare per calcolare il costo del lavoro

sostenuto dal veicolo, in termini di tempo (equivale alla variabile ct_k descritta nel paragrafo 4.2).

- **Costo Uscita [€]:** è il costo legato ad ogni mezzo, da sostenere solamente nel caso in cui il veicolo lasci il deposito per iniziare il percorso assegnato (equivale alla variabile c_k descritta nel paragrafo 4.2).
- **Capacità Massima [Kg]:** è la quantità espressa in chilogrammi di pelli che può essere caricata al massimo in uno specifico veicolo (equivale alla variabile q_k descritta nel paragrafo 4.2).
- **Costo Unitario Attesa [€/h]:** è espresso in quantità di denaro nell'unità di tempo e viene preso in considerazione ogni volta che il mezzo ritarda l'uscita dal deposito, in quanto, ad esso conviene giungere al primo macello visitato in coincidenza o successivamente all'apertura dello stesso (equivale alla variabile cr_k descritta nel paragrafo 4.2).
- **Tempo Minimo Partenza [h]:** indica l'orario di apertura del deposito, ossia il limite inferiore della finestra temporale ad esso associata; è il momento a partire dal quale il mezzo può uscire dal deposito per iniziare il tragitto assegnato (equivale alla variabile sm_k descritta nel paragrafo 4.2).
- **Tempo Massimo Partenza [h]:** indica l'orario di chiusura del deposito, ossia il limite superiore della finestra temporale ad esso associata; è il momento oltre il quale il mezzo non può più uscire dal deposito (equivale alla variabile fm_k descritta nel paragrafo 4.2).

	Costo Unitario Tempo [€/h]	Costo Uscita [€]	Capacità Massima [Kg]	Costo Unitario Attesa [€/h]	Tempo Minimo Partenza [h]	Tempo Massimo Partenza [h]
Camion_1	0,1	200	3700	0,05	10	100
Camion_2	0,1	220	2700	0,05	20	130
Camion_3	0,1	180	2700	0,05	30	120

Tabella 7.2 Inserimento dati dei camion

Come nel caso precedente, l'utente inserisce e conferma le informazioni riguardanti i veicoli, le quali, dopo aver superato i controlli previsti, vengono salvate in un array chiamato *ListaCamion()* di dimensioni n_camion e di tipo *camion* come indicato nell'Appendice relativa alle dichiarazioni delle variabili, sezione A.1.1.

7.1.3. Dati riguardanti i Tempi di carico e scarico

In tabella 7.3 sono riportati i dati di input relativi al tempo, associato ad ogni camion, necessario ad effettuare il carico di pelli presso i macelli (equivale alla variabile tt_{ik} descritta nel paragrafo 4.2) e lo scarico delle stesse in conceria (equivale alla variabile tt_{jk} descritta nel paragrafo 4.2).

	Camion_1	Camion_2	Camion_3
Deposito	0	0	0
Macello_1	20	22	24
Macello_2	25	28	30
Macello_3	32	35	37
Macello_4	18	19	20
Macello_5	30	34	36
Macello_6	22	26	28
Macello_7	19	21	22
Macello_8	41	42	44
Macello_9	23	24	26
Macello_10	44	46	47
Macello_11	32	34	35
Macello_12	29	31	32
Macello_13	34	35	37
Conceria	10	10	10

Tabella 7.3 Inserimento Tempi di carico/scarico

Le informazioni, inserite e confermate, dopo aver superato i controlli previsti, vengono salvate in una matrice chiamata *TempoCarico()* di dimensioni $(n_macelli+2, n_camion)$ e di tipo *double* come indicato nell'Appendice relativa alle dichiarazioni delle variabili, sezione A.1.1.

7.1.4. Dati riguardanti i Costi di carico e scarico

In tabella 7.4 sono riportati i dati di input relativi al costo, associato ad ogni camion, generato e sostenuto per effettuare il carico di pelli presso i macelli (equivale alla variabile dt_{ik} descritta nel paragrafo 4.2) e lo scarico delle stesse in conceria (equivale alla variabile dt_{jk} descritta nel paragrafo 4.2).

Analogamente al caso precedente, le informazioni, inserite e confermate, dopo aver superato i controlli previsti, vengono salvate in una matrice chiamata *CostoFissoCarico()* di dimensioni $(n_macelli+2, n_camion)$ e di tipo *double* come indicato nell'Appendice relativa alle dichiarazioni delle variabili, sezione A.1.1.

	Camion_1	Camion_2	Camion_3
Deposito	0	0	0
Macello_1	2	2	2
Macello_2	3	3	3
Macello_3	3	4	4
Macello_4	2	2	2
Macello_5	3	3	4
Macello_6	2	3	3
Macello_7	2	2	3
Macello_8	4	4	4
Macello_9	2	3	4
Macello_10	4	3	3
Macello_11	3	3	3
Macello_12	4	4	4
Macello_13	4	3	3
Conceria	5	6	4

Tabella 7.4 Inserimento Costi di carico/scarico

7.1.5. Dati riguardanti i Tempi di viaggio per ogni camion

In tabella 7.5 sono riportati i dati di input associati al primo camion e relativi al tempo per la percorrenza di una specifica tratta. In particolare, se nella prima riga sono riportati, per il veicolo in esame, i tempi di viaggio tra il deposito ed i vari macelli, nelle rimanenti sono indicati quelli tra un macello ed un altro e tra lo stesso e la conceria. Questi valori sono fondamentali nella fase di verifica del rispetto dei vincoli temporali e costituiscono una voce importante per il calcolo del costo di viaggio per andare da un nodo all'altro, in termini di tempo speso per effettuare lo spostamento.

Come anticipato, in tabella 7.5 sono riportati, a titolo di esempio, i dati di input associati al primo camion; all'utente viene chiesto l'inserimento di tutti i tempi di viaggio per ogni mezzo e conseguentemente si trova ad imputare tali valori in un numero di tabelle pari a n_camion . Queste informazioni, una volta inserite e confermate, dopo aver superato i controlli previsti, vengono salvate in una matrice chiamata *TempoViaggio()* di tre dimensioni equivalenti a $(n_camion, n_macelli+2, n_macelli+2)$ e di tipo *double* come indicato nell'Appendice relativa alle dichiarazioni delle variabili, sezione A.1.1. Tutti i

dati di input relativi ai tempi di viaggio per ogni camion, considerati in questo elaborato, sono riportati in Appendice, sezione A.2.1.1.

CAMION_1	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10	Macello_11	Macello_12	Macello_13	Conceria
Deposito	0	125	137	151	134	129	120	90	112	106	149	120	123	141	0
Macello_1	0	0	145	161	156	122	125	111	137	108	149	166	139	122	160
Macello_2	0	142	0	150	160	167	156	163	123	175	141	111	157	152	150
Macello_3	0	158	149	0	168	168	141	128	181	148	124	154	146	139	143
Macello_4	0	166	167	158	0	171	152	181	114	166	116	139	112	148	130
Macello_5	0	156	181	145	132	0	120	100	161	173	146	182	151	116	165
Macello_6	0	112	107	119	137	113	0	115	101	150	161	143	138	144	164
Macello_7	0	109	111	133	112	100	128	0	98	146	151	105	167	150	141
Macello_8	0	131	140	129	107	119	170	159	0	121	177	133	149	101	126
Macello_9	0	103	177	145	170	138	152	149	119	0	133	126	160	175	156
Macello_10	0	152	138	120	115	143	166	153	174	131	0	168	123	169	149
Macello_11	0	168	113	157	136	184	147	107	129	130	124	0	157	160	170
Macello_12	0	134	160	144	115	148	136	172	151	164	170	154	0	120	107
Macello_13	0	154	137	178	151	118	146	149	103	172	169	158	121	0	139
Conceria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabella 7.5 Inserimento Tempi di viaggio per Camion_1

7.1.6. Dati riguardanti i Costi di viaggio per ogni camion

In tabella 7.6 sono riportati i dati di input associati al primo camion e relativi al costo sostenuto per la percorrenza di una specifica tratta. In particolare, se nella prima riga sono riportati, per il veicolo in esame, i costi di viaggio tra il deposito ed i vari macelli, nelle rimanenti sono indicati quelli tra un macello ed un altro e tra lo stesso e la conceria. Questi valori costituiscono una voce importante per il calcolo del costo di viaggio per andare da un nodo all'altro, in termini di costo speso per effettuare lo spostamento.

Analogamente al caso precedente, in tabella 7.6 sono riportati, a titolo di esempio, i dati di input associati al primo camion; all'utente viene chiesto l'inserimento di tutti i costi di viaggio per ogni mezzo e conseguentemente si trova ad imputare tali valori in un numero di tabelle pari a n_camion . Queste informazioni, una volta inserite e confermate, dopo aver superato i controlli previsti, vengono salvate in una matrice chiamata *CostoFissoViaggio()* di tre dimensioni equivalenti a $(n_camion, n_macelli+2, n_macelli+2)$ e di tipo *double* come indicato nell'Appendice relativa alle dichiarazioni delle variabili, sezione A.1.1. Tutti i dati di input relativi ai costi di viaggio per ogni camion, considerati in questo elaborato, sono riportati in Appendice, sezione A.2.1.2.

CAMION_1	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10	Macello_11	Macello_12	Macello_13	Conceria
Deposito	0	25	37	51	34	41	33	28	31	33	29	40	35	32	0
Macello_1	0	0	45	61	56	49	51	46	39	36	48	35	51	46	41
Macello_2	0	42	0	50	60	56	60	51	57	63	47	55	44	51	39
Macello_3	0	58	49	0	68	64	51	60	42	40	31	55	50	67	50
Macello_4	0	66	67	58	0	71	66	72	61	66	43	50	31	44	45
Macello_5	0	45	58	49	58	0	61	59	77	44	53	69	52	36	41
Macello_6	0	59	71	49	61	80	0	67	63	52	75	40	62	45	35
Macello_7	0	75	65	49	51	49	39	0	29	63	42	42	76	38	37
Macello_8	0	53	45	71	47	69	70	54	0	36	64	54	57	38	26
Macello_9	0	35	65	42	67	43	54	61	38	0	51	40	57	72	29
Macello_10	0	49	46	30	42	55	73	44	65	52	0	63	45	72	31
Macello_11	0	33	74	56	49	68	39	41	54	41	64	0	52	55	27
Macello_12	0	52	42	49	32	53	63	77	56	55	42	56	0	30	37
Macello_13	0	45	53	68	42	35	46	39	37	71	74	57	29	0	40
Conceria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabella 7.6 Inserimento Costi di viaggio per Camion_1

Questa è la fase conclusiva della sezione dedicata all'inserimento dei dati di input del problema che, necessariamente, devono essere memorizzati all'interno di specifiche variabili, utilizzabili poi dall'algoritmo di calcolo. Tali variabili, come indicato precedentemente, possono essere di tipologia standard, da subito a disposizione dell'applicativo, o di tipologia personalizzata, quindi creata ad hoc per la registrazione del dato particolare in questione. A titolo di esempio, di seguito viene riportato il codice VBA per la creazione della tipologia *macello* e per la dichiarazione della variabile *ListaMacelli()* abbinata a tale tipologia e viste nel paragrafo 7.1.1.

Type macello

id As String

richiesta As Double

tempoMaxScarico As Double

apertura As Double

chiusura As Double

End Type

Public ListaMacelli() As macello

Il dimensionamento corretto di questa variabile avviene mediante l'istruzione *ReDim ListaMacelli(n_macelli + 1)* che ha l'effetto di creare un array di $(n_macelli+2)$ elementi, indicizzato da 0 a $n_macelli + 1$.

La dichiarazione di tutte le variabili, standard e personalizzate, impiegate dall'applicativo durante l'elaborazione di calcolo, è riportata in Appendice, sezione A.1.1.

7.2.Applicazione algoritmo Cluster

Le informazioni, riguardanti l'insieme dei macelli, dei veicoli disponibili nel deposito, dei costi e dei tempi di viaggio tra i nodi, dei tempi di servizio, dei costi e dei tempi di carico/scarico e dei dati sulle finestre temporali, diventano input della procedura per raggruppare tutti i clienti in cluster ammissibili e poter quindi ridurre la dimensione del problema originale. Per ottenere questo risultato, l'algoritmo di clusterizzazione, descritto nel paragrafo 5.2.1, viene puntualmente implementato in linguaggio di programmazione (il relativo codice VBA è riportato in Appendice, sezione A.1.5).

Di seguito vengono descritti i passaggi principali della procedura e, per facilitarne la comprensione, vengono mantenute le stesse suddivisioni in *Passi*, coerentemente a quanto già esposto nel paragrafo 5.2.1 ed a quanto riportato in Appendice, sezione A.1.5.

- **Passo_1a** : le informazioni riguardanti i macelli sono caricate nell'array $L()$; gli elementi di tipo *macello*, contenuti al suo interno, sono disposti, per valori crescenti dei limiti inferiori delle finestre temporali, usando il criterio di ordinamento di tipo *Bubble Sort*.
- **Passo_1b** : le informazioni riguardanti i veicoli sono caricate nell'array $V()$; gli elementi di tipo *camion*, contenuti al suo interno, sono disposti, per valori decrescenti del rapporto tra $V(j).capacita$ e $V(j).costoUscita$, usando il criterio di ordinamento di tipo *Bubble Sort* e avendo indicato con j un mezzo generico dell'array $V()$.
- **Passo_1c** : tre cicli *For* annidati ed una condizione *If* permettono l'analisi di tutti gli elementi contenuti nella matrice $CostoFissoViaggio(k, i, j)$ per poter definire $DMax$, la distanza massima consentita tra due nodi appartenenti allo stesso cluster (l'algoritmo è stato adattato allo specifico caso, in quanto non si conosce la collocazione geografica del deposito, dei macelli e della conceria, pertanto

sono stati presi in considerazione i costi di viaggio da un macello ad un altro). Analogamente viene definito il tempo *Delta* di attesa massimo consentito, passando in rassegna tutte le possibili terne *macello i – macello j – camion k* e calcolando per ciascuna di queste la differenza tra *ListaMacelli(j).apertura* e la somma tra *ListaMacelli(i).chiusura* e *TempoViaggio(k, i, j)*.

- **Passo_2** : viene creato l'array vuoto *Kn()* di tipo *macello* destinato a contenere i macelli del cluster in costruzione, abbinato al primo veicolo dell'array *V()*, avente indice *indCam* nell'array *ListaCamion()*.

La dimensione dell'array *V()* viene ridotta di una unità mediante l'istruzione *ReDim Preserve V(UBound(V) - 1)*.

- **Passo_3a** : il primo macello dell'array *L()* viene inserito nell'array *Kn()* e vengono aggiornati i seguenti parametri del cluster:
 - ✓ *aC*, limite inferiore della finestra temporale associata al cluster;
 - ✓ *bC*, limite superiore della finestra temporale associata al cluster;
 - ✓ *wC*, richiesta del cluster (somma dei fabbisogni dei clienti che lo compongono);
 - ✓ *stC*, tempo di servizio presso il cluster.

La variabile *contMac* funge da contatore per i macelli caricati all'interno del cluster abbinato all'array *Kn()*.

- **Passo_3b** : l'array *L()* è ridotto di una dimensione e viene fatta una copia dello stesso array, chiamata *Lprimo()*.
- **Passo_4** : si verifica che il carico corrente del cluster, in fase di creazione, non superi la capacità del camion, con questa condizione:
If (Lprimo(0).richiesta + wC > ListaCamion(indCam).capacita) Then ...
- **Passo_5a** : viene trovato all'interno della matrice *CostoFissoViaggio()* il costo di viaggio più basso (*dji*) tra il nodo *indMac* candidato all'inserimento nel cluster *Kn()* e tutti gli elementi contenuti nel cluster stesso.
- **Passo_5b** : viene controllato che *dji* sia inferiore a *DMax*, calcolato in precedenza al *Passo_1c*; in caso di esito negativo, attraverso il ciclo *Do ...*

Loop Until ((uscita_dji = 1) Or (uscita_Lprimo = 1)), la procedura riprende dal *Passo_4*.

- **Passo_6** : viene verificato il vincolo legato al limite superiore della finestra temporale del cluster,

If (aC + stC + *TempoViaggio*(indCam, ind_i, ind_j) <= Max_bCbj) *Then* ...

In caso di esito negativo , attraverso il ciclo *Do ... Loop Until* ((uscita_bC = 1) Or (uscita_Lprimo = 1)), la procedura riprende dal *Passo_4*.

- **Passo_7** : viene verificato il vincolo legato al limite inferiore della finestra temporale del cluster,

If (aC + stC + *TempoViaggio*(indCam, ind_i, ind_j) + Delta >= *ListaMacelli*(ind_j).apertura) *Then* ...

In caso di esito negativo , prima che la procedura riprenda dal *Passo_2*, viene chiuso il cluster avente indice *indCluster* ed eseguite in sequenza le seguenti operazioni:

- ✓ nell'array *cluster_m()*, in posizione *indCluster*, vengono salvati i campi identificativi *id* di tutti i macelli contenuti in *Kn()*;
- ✓ nell'array *nelemxcluster()*, in posizione *indCluster*, viene memorizzato il numero degli elementi *contMac* che compongono il cluster appena chiuso;
- ✓ nell'array *cluster_v()*, in posizione *indCluster*, viene salvato il campo identificativo *id* del veicolo abbinato a *Kn()*.

- **Passo_8a** : il macello *ind_j*, dopo aver superato i precedenti vincoli, viene aggiunto all'array *Kn()* con l'istruzione *Kn(contMac) = ListaMacelli(ind_j)* e vengono aggiornati i parametri *wC* (richiesta del cluster) e *stC* (tempo di servizio presso il cluster).

- **Passo_8b** : viene aggiornato il limite superiore *bC* della finestra temporale del cluster avente indice *indCluster* e l'ultimo macello inserito viene eliminato dall'array *L()* e dall'array *Lprimo()*.

- **Passo_9** : se l'array *Lprimo()* è vuoto, viene chiuso il cluster eseguendo le stesse operazioni riportate al precedente *Passo_7*, altrimenti mediante l'istruzione *GoTo PrimoNodoJdaLprimo* la procedura riprende dal *Passo_4*.

- **Passo_10** : viene ripetuta la sequenza dal *Passo_2* al *Passo_9* finchè l'array *L()* non diventa vuoto; ciò avviene con l'istruzione *Do ... Loop Until (uscita_L = 1) Or (uscita_V = 1)*. L'ultima condizione (*uscita_V = 1*) diventa attiva quando la capacità complessiva della flotta dei veicoli risulta essere inferiore alla totalità della domanda dei clienti.

L'applicazione di questo algoritmo al caso oggetto di studio, porta all'individuazione di due raggruppamenti così definiti:

- *Cluster_1*: *Macello_1, Macello_5, Macello_6, Macello_7, Macello_9, Macello_11, Macello_12, Macello_13*
 - Servito dal *Camion_1*
- *Cluster_2*: *Macello_2, Macello_3, Macello_4, Macello_8, Macello_10*.
 - Servito dal *Camion_3*

I dati dei clienti contenuti nei cluster, ottenuti mediante questa procedura, diventano elementi di input distinti per la fase successiva di inserzione; l'algoritmo euristico di *Cheapest Insertion*, descritto nel paragrafo successivo, definisce il sequenziamento (*route*) dei clienti contenuti all'interno di ciascun cluster, a cui è già stato assegnato il veicolo.

7.3.Applicazione algoritmo Cheapest Insertion

L'implementazione in *VBA* dell'algoritmo di *Cheapest Insertion* è divisa in cinque parti logiche, così definite:

- predisposizione delle variabili per permettere l'iterazione e l'applicazione dell'algoritmo ad ogni cluster individuato nella fase precedente;
- fase di inizializzazione applicata ad un cluster specifico;
- fase di assegnazione applicata ad un cluster specifico;
- verifica dei vincoli e visualizzazione delle informazioni del contesto (dati, tabelle) a seguito di ogni nuova possibile assegnazione;
- calcolo dei costi finali del tragitto e stampa delle informazioni di dettaglio ad esso connesse.

7.3.1.Predisposizione variabili

Per iniziare, è necessario calcolare i costi totali che insorgono nel tragitto tra ogni coppia di macelli inseriti per ogni camion a disposizione; per fare questo, vengono

create ed alimentate, con i rispettivi dati corretti, le seguenti matrici di tipo *double*, ciascuna coerente ad uno specifico concetto fra quelli già espressi nel paragrafo 6.1.

ReDim TempoAttesa(n_camion - 1, n_macelli + 1, n_macelli + 1)

ReDim TempoTotale(n_camion - 1, n_macelli + 1, n_macelli + 1)

ReDim CostoAttesa(n_camion - 1, n_macelli + 1, n_macelli + 1)

ReDim CostoTotViaggio(n_camion - 1, n_macelli + 1, n_macelli + 1)

ReDim CostoTotCarico(n_macelli + 1, n_camion - 1)

ReDim CostoTotale(n_camion - 1, n_macelli + 1, n_macelli + 1)

La procedura, attraverso la *Sub InizioCheapestInsertion()* esegue le attività seguenti:

- a. Calcola tutti gli elementi delle matrici precedentemente descritte.
- b. Aggiunge un foglio di lavoro per ogni tragitto in fase di sequenziamento, nominato come *ActiveSheet.Name = "Tragitto Num." & cont + 1*.
- c. Mediante la *Sub PrimoMacello()*, avvia la fase di inizializzazione che avrà poi il seguito dell'assegnazione e della verifica dei vincoli.
- d. Avvia il calcolo dei costi del tragitto individuato con la *Sub CalcoloCostiTragitto()*.
- e. Aggiorna la soluzione finale del problema con l'istruzione:
CostoFinaleProblema = CostoFinaleProblema + CostoFinaleTragitto.
- f. Inserisce, mediante la *Sub TabellaTragittoFinale()*, i dati riepilogativi del tragitto individuato, sia nel foglio di lavoro riportante la soluzione finale del problema (*ActiveSheet.Name = "Soluzione_Finale"*) e sia nella scheda associata al tragitto (punto b).
- g. Itera tutte le attività precedenti, tante volte quanti sono i cluster precedentemente individuati (*n_cluster*).

Il codice VBA che implementa questa parte dell'algoritmo è riportato in Appendice, sezione A.1.6.1.

7.3.2.Fase di inizializzazione applicata ad un cluster specifico

La procedura, mediante la *Sub PrimoMacello()* esegue le seguenti operazioni:

- a. Memorizza l'indice del cluster in esame all'interno della variabile *indCluster*.
- b. Crea l'array *tragitto()* destinato poi a contenere elementi personalizzati di tipo *MacelloAssegnato* (come indicato in Appendice, sezione A.1.1), ossia i

componenti del route in fase di costruzione; il primo nodo del *tragitto()* è il deposito e l'ultimo la conceria.

- c. Memorizza l'indice del veicolo, abbinato al tragitto in costruzione, all'interno della variabile *indCam*.
- d. Carica tutti gli indici dei macelli del cluster all'interno dell'array *MacNonAss()* destinato a contenere elementi di tipo *integer*.
- e. Ricerca all'interno della matrice *CostoTotale()* il macello che presenta il costo minimo dal deposito e lo inserisce nel *tragitto()* come primo nodo, tra il deposito e la conceria.
- f. Considera tutti gli elementi di tipo *MacelloAssegnato* contenuti in *tragitto()* e aggiorna i campi di ciascuno di essi, ossia: *capacita Residua*, *tempoCaricoAss*, *costoCaricoAss*, *tempoViaggioAss*, *costoViaggioAss*, *tempoAttesaAss* e *tempoFineServ* (questi campi sono coerenti ai concetti già esposti nel paragrafo 6.2).
- g. Visualizza mediante la *Sub TabellaTragitto()*, nel foglio di lavoro associato al percorso, la tabella del route in fase di costruzione.
- h. Avvia la fase successiva, di assegnazione, con la *Sub Assegnazione()*.

Tra il deposito e la conceria è stato inserito come primo nodo quello che, tra tutti i componenti del *Cluster_1*, presenta il minimo costo dal deposito (237,5), ossia il *Macello_1*, come evidenziato in figura 7.3, usando il *Camion_1*.

Espressione	Valore
CostoTotale	
CostoTotale(0)	
CostoTotale(0,0)	
CostoTotale(0,0,0)	0
CostoTotale(0,0,1)	237,5
CostoTotale(0,0,2)	275
CostoTotale(0,0,3)	269
CostoTotale(0,0,4)	279
CostoTotale(0,0,5)	262
CostoTotale(0,0,6)	283
CostoTotale(0,0,7)	316
CostoTotale(0,0,8)	290
CostoTotale(0,0,9)	244
CostoTotale(0,0,10)	301
CostoTotale(0,0,11)	252
CostoTotale(0,0,12)	261
CostoTotale(0,0,13)	246,1
CostoTotale(0,0,14)	242

Figura 7.3 Costo Totale minimo per inserzione primo macello

Il tragitto parziale dopo la fase di inizializzazione è riportato nella tabella 7.7.

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Conceria	3450	40	285	105	430	7	66

Tabella 7.7 Tragitto dopo la fase di inizializzazione

Il codice VBA che implementa questa parte dell'algoritmo è riportato in Appendice, sezione A.1.6.2.

7.3.3. Fase di assegnazione applicata ad un cluster specifico

Il sequenziamento dei nodi di ogni cluster viene creato in maniera iterativa e gestito con l'array *tragitto()* che viene ridimensionato ad ogni nuovo inserimento.

L'algoritmo parte ora con l'analizzare ogni coppia di macelli adiacenti già presenti nel percorso per stabilire quale, fra i nodi non ancora assegnati e quindi contenuti in *MacNonAss()*, è possibile includere in modo da generare il minimo costo di inserzione. Se *indMacPrec* e *indMacSuc* identificano rispettivamente gli indici del primo e del secondo macello della coppia in esame, *indMac* indica un macello non ancora assegnato e *indCam* identifica il veicolo impiegato per il servizio, allora, coerentemente con quanto già esposto nel paragrafo 6.2, il costo di inserzione viene calcolato come segue:

CostiGenerati(j).costo =

CostoTotale(indCam, indMacPrec, indMac) +

CostoTotale(indCam, indMac, indMacSuc) –

CostoTotale(indCam, indMacPrec, indMacSuc).

Per ogni coppia di nodi adiacenti del percorso viene creato un array chiamato *CostiGenerati()*, contenente tutti i macelli candidati all'assegnazione con i rispettivi costi di inserzione ordinati secondo criterio crescente. Questo vettore contiene elementi di tipo *costoGenerato* (vedi Appendice, sezione A.1.1) ed in base all'ordinamento subito, riporta in prima posizione (ad indice zero) il macello con il costo di inserzione minore per la coppia di nodi adiacenti a cui l'array è abbinato. Ogni posizione del vettore *CostiGenerati()* è riconducibile direttamente ad un elemento

contenuto nell'array *MacNonAss()* ed entrambe queste strutture dati hanno lo stesso numero di componenti. I concetti appena esposti, applicati al *tragitto()* subito dopo la fase di inizializzazione, sono rappresentati in figura 7.4.

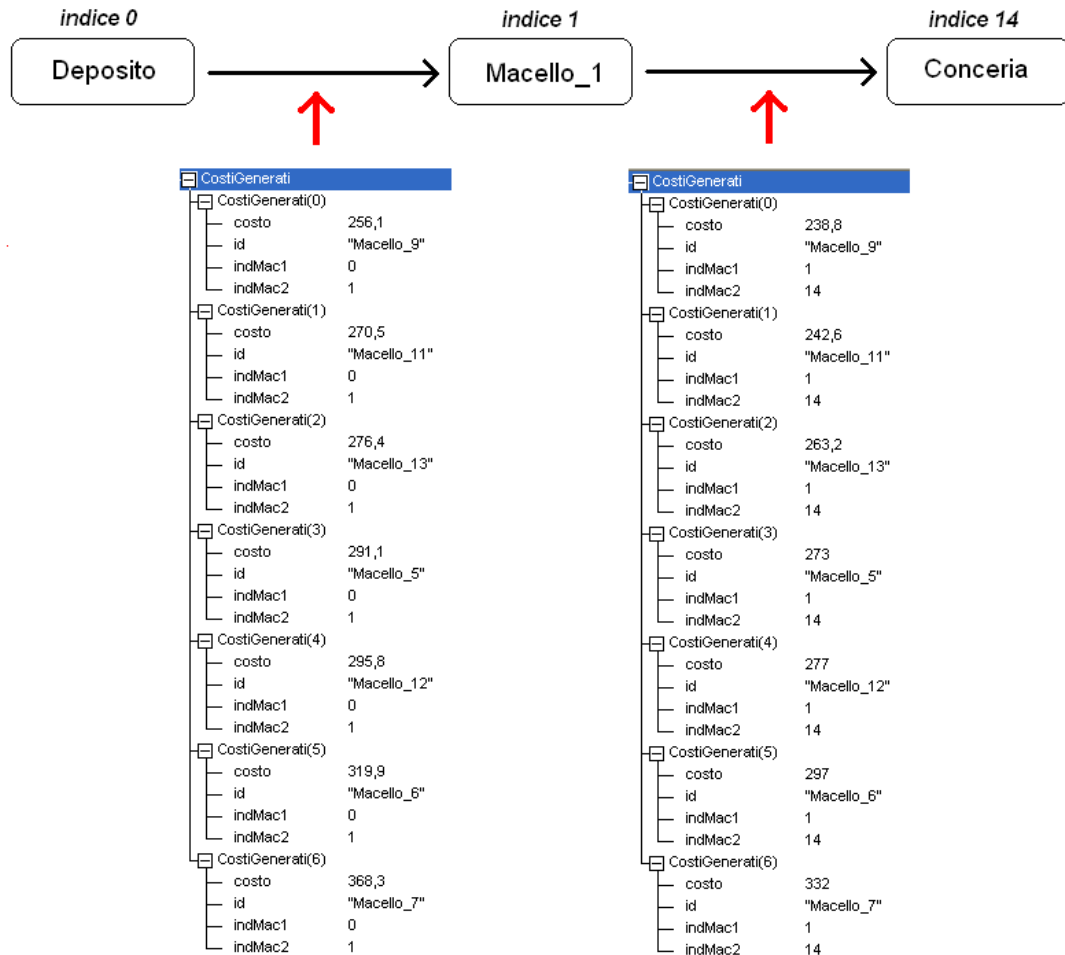


Figura 7.4 Rappresentazione dei vettori *CostiGenerati()* usati nell'algoritmo

Il primo elemento dell'array *CostiGenerati()* di sinistra è il *Macello_9* e genera un costo di inserzione di 256,1 se inserito tra il *Deposito* (*indMac1=0*) ed il *Macello_1* (*indMac2=1*); il primo elemento dell'array *CostiGenerati()* di destra è sempre il *Macello_9*, ma questa volta genera un costo di inserzione di 238,8 se inserito tra il *Macello_1* (*indMac1=1*) ed la *Conceria* (*indMac2=14*).

I diversi vettori *CostiGenerati()*, creati per ogni coppia di nodi adiacenti del percorso, vanno a costituire, di volta in volta, le colonne della matrice *CostiGeneratiTot()*; essa contiene i costi di inserzione che tutti i macelli non ancora assegnati (righe della matrice) generano per tutte le coppie di macelli presenti nel tragitto (colonne della matrice). Ad ogni iterazione, a mano a mano che si procede con gli inserimenti,

verranno create matrici *CostiGeneratiTot()* con un numero di colonne sempre maggiore ed un numero di righe sempre minore. Gli elementi all'interno di tali matrici sono di tipo *costoGenerato* e distinguibili univocamente in base al valore dei loro quattro campi: *id*, *costo*, *indMac1* e *indMac2*.

L'algoritmo ricerca, all'interno della matrice *CostiGeneratiTot()*, il costo minimo di inserzione e per tale valore individua il macello responsabile e gli indici dei nodi della coppia a cui fa riferimento. Nel caso oggetto di studio, come si evince dalla figura 7.4, il minimo costo di inserzione è di 238,8 e quindi l'algoritmo, nei passi successivi, proverà ad inserire il *Macello_9* tra il *Macello_1* e la *Conceria*. Tutti i macelli in *CostiGeneratiTot()* che di volta in volta si esaminano per una eventuale assegnazione, vengono inseriti nell'array *MacEsaminati()* in modo che, qualora non superassero i vincoli per poter essere inseriti nel *tragitto()*, non possano più essere presi in considerazione durante la ricerca di un nuovo minimo.

Proseguendo con l'esempio, il *Macello_9* viene inserito come terzo elemento nell'array *tragitto()*, tra il *Macello_1* e la *Conceria*; tale aggiunta, prima della verifica dei vincoli, richiede l'aggiornamento di tutti i campi degli elementi contenuti nel *tragitto()*, come indicato nel paragrafo 7.3.2 al punto f. In questo caso, l'esito del rispetto dei vincoli è positivo, quindi il *Macello_9* viene assegnato al *tragitto()* come riportato in tabella 7.8.

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Conceria	2970	63	389	0	452	9	90

Tabella 7.8 Tragitto dopo l'inserimento del *Macello_9*

Alla prossima iterazione, tutto il procedimento viene rieseguito considerando una coppia di macelli in più nel percorso: *Deposito – Macello_1, Macello_1 – Macello_9, Macello_9 – Conceria*. Vengono passati in rassegna tutti i macelli presenti in *MacNonAss()* e calcolati i nuovi costi di inserzione, costruendo una matrice *CostiGeneratiTot()* con una dimensione in più per quanto riguarda le colonne (coppie di macelli del percorso) ed una in meno per quanto riguarda le righe (macelli non ancora assegnati). Viene inserito il nuovo macello trovato e ricontrollati tutti i vincoli per

confermare o meno tale assegnazione. In caso di esito negativo, non si può procedere con l'assegnazione, il macello viene tolto dal *tragitto()* e l'algoritmo, grazie all'istruzione *GoTo RicercaCostoInserimentoMinimo*, esegue la ricerca dell'elemento avente il secondo minimo di inserzione all'interno della matrice *CostiGeneratiTot()*.

Dopo ogni assegnazione, prima di eseguire l'istruzione *GoTo NuovaAssegnazione* per considerare i successivi candidati ad un ulteriore inserimento, è importante aggiornare il costo di attesa per tutti i macelli, dal momento che un inserimento va ad alterare i tempi di fine servizio, come spiegato nel paragrafo 6.2.

L'algoritmo a questo punto si ripete fino a quando tutti i macelli sono stati assegnati oppure fino a quando non è più possibile assegnare macelli che rispettino tutti i vincoli per un determinato veicolo. In quest'ultimo caso, gli elementi che rimangono all'interno dell'array *MacNonAss()* vengono inseriti nel cluster successivo per diventare poi oggetto di analisi dell'algoritmo di *Cheapest Insertion* ad esso applicato. La procedura esegue in sequenza la *Sub CalcoloCostiTragitto()*, per calcolare i costi del percorso appena individuato, e la *Sub TabellaTragittoFinale()*, per visualizzare i dati finali del percorso nel rispettivo foglio di lavoro e nella scheda riepilogativa chiamata "Soluzione_Finale".

Il codice VBA che implementa questa parte dell'algoritmo è riportato in Appendice, sezione A.1.6.3.

7.3.4.Verifica dei vincoli e visualizzazione dati di elaborazione

Come già anticipato, ciò che conferma o meno un'assegnazione è il superamento dei vincoli con esito positivo; mediante il richiamo della *Function verifica_vincoli()* avviene, dopo ogni nuovo inserimento, il controllo e la verifica delle condizioni restrittive imposte dal problema in tutti i macelli già assegnati al percorso in costruzione. Coerentemente con quanto già descritto nel paragrafo 6.3, se ad esempio *q* indica la posizione di un macello all'interno del *tragitto()* ed *im* il suo indice per identificarlo all'interno dell'array *ListaMacelli()*, i vincoli sono implementati in VBA secondo queste tre espressioni:

- Vincolo di capacità
If (tragitto(q).capacitaResidua >= 0) Then ...
- Vincolo di tempo di chiusura del macello
If (tragitto(q).tempoFineServ <= ListaMacelli(im).chiusura) Then ...
- Vincolo di tempo massimo di scarico
If (tragitto(q).tempoFineServ <= ListaMacelli(im).tempoMaxScarico) Then...

Queste condizioni restrittive devono chiaramente valere per ciascun elemento inserito all'interno del percorso e, per ottenere ciò, il controllo viene iterato grazie a questa istruzione VBA: *For q = 1 To UBound(tragitto) ... Next q*. Al termine di questo ciclo, affinché l'assegnazione possa andare a buon fine (*verifica_vincoli = True*), i controlli con esito positivo devono essere pari a tre volte il numero dei nodi inseriti nel *tragitto()*, con l'esclusione della posizione zero (deposito). Questo dato viene monitorato dalla variabile *totVincoliOK* che incrementa il suo valore di un'unità quando in un macello tutti e tre i vincoli (*I_Vincolo*, *II_Vincolo*, *III_Vincolo*) assumono il valore *True*. Alla fine, per certificare positivamente l'assegnazione, deve valere questa condizione: *totVincoliOK = UBound(tragitto)*. In questo caso, la procedura esegue la *Sub TabellaTragitto()* che visualizza, nel foglio di lavoro associato al percorso, tutti i valori dei campi dei macelli contenuti in *tragitto()*, come riporta l'esempio in tabella 7.9 relativo alla quinta iterazione di assegnazione.

Nel tragitto è stato aggiunto il Macello_12 tra il Macello_13 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_6	2230	107	506	0	613	9	140
Macello_13	1960	141	650	0	791	13	185
Macello_12	1410	170	771	0	941	17	214
Conceria	1410	180	878	0	1058	22	251

Tabella 7.9 Tragitto dopo il sesto inserimento

Nel caso contrario, di non superamento dei vincoli, l'algoritmo esegue la *Sub StampaRigaNonInserimento()* che visualizza, sempre nel foglio associato al percorso, la prima motivazione riscontrata per la quale non è stato possibile confermare l'assegnazione, come riportato in tabella 7.10 alla sesta iterazione.

Nel tragitto NON è stato possibile aggiungere il Macello_5 tra il Macello_12 e la Conceria

Il tempo di fine servizio (1119) presso il Macello_5 è superiore al tempo di chiusura dello stesso (900)

Tabella 7.10 Esito negativo dell'assegnazione alla sesta iterazione

Il codice VBA che implementa questa parte dell'algoritmo è riportato in Appendice, sezione A.1.6.4.

Per ciascuno dei due cluster precedentemente individuati (vedi paragrafo 7.2), è stato trovato il tragitto ottimale che soddisfa le richieste dei clienti; le informazioni che descrivono i processi intermedi di elaborazione dati sono riportate in Appendice, sezione A.2.2.1 per il tragitto numero 1 e sezione A.2.2.2 per il tragitto numero 2.

7.3.5. Calcolo costi finali del tragitto ottenuto in un cluster specifico

Una volta completata la fase di assegnazione e verifica dei vincoli per un determinato cluster, l'algoritmo esegue la *Sub CalcoloCostiTragitto()* per determinare i costi di ciascun percorso individuato. Le operazioni svolte da questa *routine* sono coerenti ai concetti affrontati in precedenza nel paragrafo 6.4. In modo particolare, dopo la determinazione del costo d'attesa per il primo macello visitato (*CostoAttesaPrimoMac*) e del costo del tempo di viaggio totale (*CostoTempoTragitto*), il costo finale del tragitto viene calcolato con la seguente espressione in codice VBA:

CostoFinaleTragitto =

$$\begin{aligned} & \text{ListaCamion}(\text{indCam}).\text{costoUscita} + _ \\ & \text{tragitto}(\text{UBound}(\text{tragitto})).\text{costoViaggioAss} + _ \\ & \text{tragitto}(\text{UBound}(\text{tragitto})).\text{costoCaricoAss} + _ \\ & \text{CostoAttesaPrimoMac} + _ \\ & \text{CostoTempoTragitto} \end{aligned}$$

L'indice valorizzato con *UBound(tragitto)* identifica il nodo finale (*Conceria*) mentre *indCam* è la posizione del veicolo, impiegato nel servizio di raccolta/consegna, all'interno dell'array *ListaCamion()*.

La procedura mediante la *Sub TabellaTragittoFinale()* visualizza, sia nel foglio di lavoro associato al percorso che nella scheda riepilogativa finale, i dati del route individuato, contenuti all'interno dell'array *tragitto()*. Nelle tabelle 7.11 e 7.12 sono riportati i costi finali, rispettivamente, del tragitto numero 1 e del tragitto numero 2.

- **Tragitto numero 1:** *Deposito - Macello_1 - Macello_5 - Macello_9 - Macello_11 - Macello_6 - Macello_13 - Macello_12 - Macello_7 - Conceria*
 - Servito dal *Camion_1*

TRAGITTO FINALE del Camion_1

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_5	3060	60	247	0	307	5	74
Macello_9	2580	83	420	0	503	7	118
Macello_11	2180	115	546	0	661	10	158
Macello_6	1840	137	693	0	830	12	197
Macello_13	1570	171	837	0	1008	16	242
Macello_12	1020	200	958	0	1158	20	271
Macello_7	600	219	1130	0	1349	22	348
Conceria	600	229	1271	0	1500	27	385

TRAGITTO NUMERO 1: costo totale del percorso associato al Camion_1

761

Tabella 7.11 Costo finale tragitto numero 1

- **Tragitto numero 2:** Deposito - Macello_3 - Macello_2 - Macello_4 - Macello_8 - Macello_10 - Conceria
 - Servito dal Camion_3

TRAGITTO FINALE del Camion_3

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Macello_4	1070	117	471	3	591	9	164
Macello_8	680	161	586	3	750	13	225
Macello_10	60	208	764	3	975	16	287
Conceria	60	218	896	3	1117	20	331

TRAGITTO NUMERO 2: costo totale del percorso associato al Camion_3

639.7

Tabella 7.12 Costo finale tragitto numero 2

Il codice VBA che implementa questa parte dell'algoritmo è riportato in Appendice, sezione A.1.6.5.

7.4. Soluzione finale del problema

La soluzione conclusiva, ossia il costo finale totale del problema oggetto di questo caso studio, è costituita dalla somma dei costi finali di ogni singolo tragitto individuato dalla procedura. La somma di tutte le variabili *CostoFinaleTragitto*, calcolate per ogni percorso individuato dall'algoritmo, fornisce la soluzione e quindi il costo finale del problema. Per ottenerlo, l'espressione seguente in codice VBA viene iterata tante volte (*cont*) quanti sono i cluster (*n_cluster*) precedentemente individuati.

$$\text{CostoFinaleProblema} = \text{CostoFinaleProblema} + \text{CostoFinaleTragitto}$$

In tabella 7.13 è riportato il costo finale del problema (1400.7), dato dalla somma del costo finale tragitto numero 1 (761) e del costo finale tragitto numero 2 (639.7), a cui si giunge dopo un tempo di elaborazione di circa 20 secondi.

COSTO TOTALE FINALE RELATIVO AL PROBLEMA DI ROUTING	1400.7
--	---------------

Tabella 7.13 Costo finale del problema

Il codice VBA che implementa questa parte dell'algoritmo è riportato in Appendice, sezione A.1.6.5.

7.5. Prova con 20 macelli e 5 camion

L'analisi del problema viene ora estesa ad un contesto pratico costituito da 20 macelli e 5 camion per testare le potenzialità dell'algoritmo sviluppato; nei paragrafi seguenti è descritto in modo sequenziale lo svolgimento della procedura applicata a questa situazione reale.

7.5.1. Inserimento dati

Come nel caso precedente, dopo aver selezionato il tasto "INIZIO" evidenziato in figura 7.1, l'utente può inserire, all'interno dell'interfaccia riportata in figura 7.2, il numero dei clienti che devono essere soddisfatti ed il numero dei veicoli a disposizione per il trasporto del materiale. Dopo questa operazione, come di seguito esposto, inizia il caricamento di tutte le informazioni necessarie all'applicativo per risolvere il problema.

7.5.1.1. Dati riguardanti i Macelli e la Conceria

I primi dati da inserire sono quelli riportati in tabella 7.14, relativi ai macelli ed alla conceria. Se la prima riga è riservata al deposito e l'ultima alla conceria, in ciascuna delle righe rimanenti sono indicate le informazioni di uno specifico macello; gli elementi che identificano le quattro colonne sono spiegati nel paragrafo 7.1.1.

	Richiesta [Kg]	Tempo Max Scarico [h]	Apertura [h]	Chiusura [h]
Deposito	0	0	0	0
Macello_1	250	1700	120	540
Macello_2	530	1800	380	960
Macello_3	800	1800	180	490
Macello_4	300	1900	450	840
Macello_5	390	1701	210	900
Macello_6	340	1600	500	1500
Macello_7	420	1900	880	1400
Macello_8	390	1500	590	1200
Macello_9	480	1600	110	710
Macello_10	620	1800	720	1060
Macello_11	400	1700	100	1000
Macello_12	550	1700	260	1300
Macello_13	270	1500	80	1440
Macello_14	200	1800	310	650
Macello_15	180	1700	120	820
Macello_16	350	1500	260	740
Macello_17	420	1650	290	1400
Macello_18	380	1850	310	1100
Macello_19	510	1900	400	1200
Macello_20	280	1550	340	980
Conceria	0	0	420	1740

Tabella 7.14 Inserimento dati dei macelli e della conceria

7.5.1.2. Dati riguardanti i Camion

In tabella 7.15 sono riportati i dati di input relativi ai mezzi presenti nel deposito e disponibili per il trasporto delle merci (gli elementi identificati dalle sei colonne sono spiegati nel paragrafo 7.1.2).

	Costo Unitario Tempo [€/h]	Costo Uscita [€]	Capacità Massima [Kg]	Costo Unitario Attesa [€/h]	Tempo Minimo Partenza [h]	Tempo Massimo Partenza [h]
Camion_1	0,1	200	3700	0,05	10	100
Camion_2	0,1	220	2700	0,05	20	130
Camion_3	0,1	180	2700	0,05	30	120
Camion_4	0,1	210	3700	0,05	20	110
Camion_5	0,1	200	2700	0,05	10	120

Tabella 7.15 Inserimento dati dei camion

7.5.1.3. Dati riguardanti i Tempi di carico e scarico

In tabella 7.16 sono riportati i dati di input relativi al tempo, associato ad ogni camion, necessario ad effettuare il carico di pelli presso i macelli (equivale alla variabile tt_{ik} descritta nel paragrafo 4.2) e lo scarico delle stesse in conceria (equivale alla variabile tt_{jk} descritta nel paragrafo 4.2).

	Camion_1	Camion_2	Camion_3	Camion_4	Camion_5
Deposito	0	0	0	0	0
Macello_1	20	22	24	21	23
Macello_2	25	28	30	26	29
Macello_3	32	35	37	34	36
Macello_4	18	19	20	21	17
Macello_5	30	34	36	31	33
Macello_6	22	26	28	27	21
Macello_7	19	21	22	23	20
Macello_8	41	42	44	43	45
Macello_9	23	24	26	25	22
Macello_10	44	46	47	43	45
Macello_11	32	34	35	33	36
Macello_12	29	31	32	31	30
Macello_13	34	35	37	36	35
Macello_14	25	27	28	23	25
Macello_15	35	33	36	34	31
Macello_16	26	27	25	23	24
Macello_17	41	40	43	42	40
Macello_18	31	32	29	30	28
Macello_19	32	33	33	31	32
Macello_20	18	23	22	21	19
Conceria	10	10	10	10	10

Tabella 7.16 Inserimento Tempi di carico/scarico

7.5.1.4. Dati riguardanti i Costi di carico e scarico

In tabella 7.17 sono riportati i dati di input relativi al costo, associato ad ogni camion, generato e sostenuto per effettuare il carico di pelli presso i macelli (equivale alla variabile dt_{ik} descritta nel paragrafo 4.2) e lo scarico delle stesse in conceria (equivale alla variabile dt_{jk} descritta nel paragrafo 4.2).

	Camion_1	Camion_2	Camion_3	Camion_4	Camion_5
Deposito	0	0	0	0	0
Macello_1	2	2	2	2	2
Macello_2	3	3	3	3	3
Macello_3	3	4	4	4	3
Macello_4	2	2	2	3	3
Macello_5	3	3	4	4	3
Macello_6	2	3	3	2	2
Macello_7	2	2	3	3	2
Macello_8	4	4	4	3	4
Macello_9	2	3	4	4	3
Macello_10	4	3	3	4	4
Macello_11	3	3	3	4	3
Macello_12	4	4	4	4	4
Macello_13	4	3	3	3	4
Macello_14	3	3	4	4	3
Macello_15	4	3	2	3	2
Macello_16	3	4	2	2	3
Macello_17	3	3	3	3	3
Macello_18	2	3	2	2	3
Macello_19	3	2	4	3	4
Macello_20	3	2	2	2	3
Conceria	5	6	4	5	4

Tabella 7.17 Inserimento Costi di carico/scarico

7.5.1.5. Dati riguardanti i Tempi di viaggio per ogni camion

Nelle tabelle 7.18 e 7.19 sono riportati i dati di input associati al primo camion e relativi al tempo per la percorrenza di una specifica tratta. In particolare, se nella prima riga sono riportati, per il veicolo in esame, i tempi di viaggio tra il deposito ed i vari macelli, nelle rimanenti sono indicati quelli tra un macello ed un altro e tra lo stesso e la conceria. Questi valori sono fondamentali nella fase di verifica del rispetto dei vincoli temporali e costituiscono una voce importante per il calcolo del costo di viaggio per andare da un nodo all'altro, in termini di tempo speso per effettuare lo spostamento. Tutti i dati di input relativi ai tempi di viaggio per ogni mezzo, considerati in questo elaborato, sono riportati in Appendice, sezione A.3.1.1.

CAMION_1	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
Deposito	0	125	137	151	134	129	120	90	112	106	149
Macello_1	0	0	145	161	156	122	125	111	137	108	149
Macello_2	0	142	0	150	160	167	156	163	123	175	141
Macello_3	0	158	149	0	168	168	141	128	181	148	124
Macello_4	0	166	167	158	0	171	152	181	114	166	116
Macello_5	0	156	181	145	132	0	120	100	161	173	146
Macello_6	0	112	107	119	137	113	0	115	101	150	161
Macello_7	0	109	111	133	112	100	128	0	98	146	151
Macello_8	0	131	140	129	107	119	170	159	0	121	177
Macello_9	0	103	177	145	170	138	152	149	119	0	133
Macello_10	0	152	138	120	115	143	166	153	174	131	0
Macello_11	0	168	113	157	136	184	147	107	129	130	124
Macello_12	0	134	160	144	115	148	136	172	151	164	170
Macello_13	0	154	137	178	151	118	146	149	103	172	169
Macello_14	0	109	153	121	129	144	109	153	121	129	144
Macello_15	0	111	154	166	142	119	111	154	166	142	119
Macello_16	0	177	144	115	158	154	177	144	115	158	154
Macello_17	0	151	128	159	148	141	151	128	159	148	141
Macello_18	0	170	119	141	114	150	170	119	141	114	150
Macello_19	0	175	148	186	155	119	175	148	186	155	119
Macello_20	0	153	165	148	141	146	153	165	148	141	146
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tabella 7.18 Inserimento Tempi di viaggio per Camion_1 – prima parte

CAMION_1	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
Deposito	120	123	141	134	111	154	166	142	119	112	0
Macello_1	166	139	122	128	177	144	115	158	154	140	160
Macello_2	111	157	152	139	151	128	159	148	141	149	150
Macello_3	154	146	139	153	170	119	141	114	150	117	143
Macello_4	139	112	148	139	175	148	186	155	119	145	130
Macello_5	182	151	116	141	153	165	148	141	146	151	165
Macello_6	143	138	144	128	149	153	109	170	153	102	164
Macello_7	105	167	150	158	92	120	109	153	121	176	141
Macello_8	133	149	101	160	112	139	111	154	166	170	126
Macello_9	126	160	175	136	166	125	177	144	115	161	156
Macello_10	168	123	169	124	131	187	151	128	159	121	149
Macello_11	0	157	160	151	190	116	170	119	141	148	170
Macello_12	154	0	120	144	101	170	175	148	186	165	107
Macello_13	158	121	0	137	116	107	153	165	148	146	139
Macello_14	128	126	143	0	166	142	119	131	125	112	148
Macello_15	131	125	140	138	0	132	162	158	160	151	116
Macello_16	142	139	111	144	128	0	150	140	132	138	144
Macello_17	157	125	177	119	131	120	0	139	142	167	150
Macello_18	146	187	151	154	142	148	150	0	144	149	101
Macello_19	142	116	170	141	157	160	163	168	0	160	175
Macello_20	129	170	175	150	146	162	174	140	144	0	151
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tabella 7.19 Inserimento Tempi di viaggio per Camion_1 – seconda parte

7.5.1.6. Dati riguardanti i Costi di viaggio per ogni camion

Nelle tabelle 7.20 e 7.21 sono riportati i dati di input associati al primo camion e relativi al costo sostenuto per la percorrenza di una specifica tratta. In particolare, se nella prima riga sono riportati, per il veicolo in esame, i costi di viaggio tra il deposito ed i vari macelli, nelle rimanenti sono indicati quelli tra un macello ed un altro e tra lo stesso e la conceria. Questi valori costituiscono una voce importante per il calcolo del costo di viaggio per andare da un nodo all'altro, in termini di costo speso per effettuare lo spostamento.

Analogamente al caso precedente, tutti i dati di input relativi ai costi di viaggio per ogni camion, considerati in questo elaborato, sono riportati in Appendice, sezione A.3.1.2.

CAMION_1	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
Deposito	0	25	37	51	34	41	33	28	31	33	29
Macello_1	0	0	45	61	56	49	51	46	39	36	48
Macello_2	0	42	0	50	60	56	60	51	57	63	47
Macello_3	0	58	49	0	68	64	51	60	42	40	31
Macello_4	0	66	67	58	0	71	66	72	61	66	43
Macello_5	0	45	58	49	58	0	61	59	77	44	53
Macello_6	0	59	71	49	61	80	0	67	63	52	75
Macello_7	0	75	65	49	51	49	39	0	29	63	42
Macello_8	0	53	45	71	47	69	70	54	0	36	64
Macello_9	0	35	65	42	67	43	54	61	38	0	51
Macello_10	0	49	46	30	42	55	73	44	65	52	0
Macello_11	0	33	74	56	49	68	39	41	54	41	64
Macello_12	0	52	42	49	32	53	63	77	56	55	42
Macello_13	0	45	53	68	42	35	46	39	37	71	74
Macello_14	0	44	52	49	65	60	44	23	35	40	65
Macello_15	0	66	58	71	45	38	71	38	66	44	43
Macello_16	0	41	54	64	41	43	39	34	32	63	45
Macello_17	0	24	52	55	35	41	62	30	40	50	71
Macello_18	0	32	73	28	50	60	71	42	55	30	29
Macello_19	0	41	39	72	60	51	42	36	61	69	54
Macello_20	0	39	54	44	41	25	38	67	45	37	33
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tabella 7.20 Inserimento Costi di viaggio per Camion_1 – prima parte

CAMION_1	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
Deposito	40	35	32	36	42	33	34	29	17	30	0
Macello_1	35	51	46	28	50	48	29	35	32	37	41
Macello_2	55	44	51	39	29	25	34	38	53	26	39
Macello_3	55	50	67	53	40	31	53	50	69	53	50
Macello_4	50	31	44	39	66	44	50	31	44	39	45
Macello_5	69	52	36	37	44	53	69	52	36	37	41
Macello_6	40	62	45	36	52	70	41	62	45	36	35
Macello_7	42	76	38	41	63	42	42	77	38	41	37
Macello_8	54	57	38	58	36	64	54	58	38	58	26
Macello_9	40	57	72	36	42	51	40	57	73	39	29
Macello_10	63	45	72	26	52	43	63	44	72	27	31
Macello_11	0	52	55	49	41	64	37	52	61	48	27
Macello_12	56	0	30	40	55	42	56	36	30	40	37
Macello_13	57	29	0	42	53	62	64	70	34	38	40
Macello_14	72	32	63	0	60	72	55	50	29	48	35
Macello_15	66	38	36	71	0	72	47	27	35	54	45
Macello_16	49	35	44	50	60	0	61	46	34	53	41
Macello_17	48	28	30	72	67	40	0	33	69	41	29
Macello_18	33	35	62	61	66	51	54	0	35	40	51
Macello_19	45	31	44	47	72	38	45	39	0	39	27
Macello_20	71	69	71	43	47	33	31	29	65	0	37
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tabella 7.21 Inserimento Costi di viaggio per Camion_1 – seconda parte

7.5.2. Applicazione algoritmo Cluster

Le informazioni, riguardanti l'insieme dei macelli, dei veicoli disponibili nel deposito, dei costi e dei tempi di viaggio tra i nodi, dei tempi di servizio, dei costi e dei tempi di carico/scarico e dei dati sulle finestre temporali, diventano input della procedura per raggruppare tutti i clienti in cluster ammissibili e poter quindi ridurre la dimensione del problema originale.

L'applicazione dell'algoritmo di clusterizzazione al caso oggetto di studio, porta all'individuazione di tre raggruppamenti così definiti:

- *Cluster_1: Macello_1, Macello_6, Macello_9, Macello_11, Macello_12, Macello_13, Macello_15, Macello_17.*
 - *Servito dal Camion_1*

- *Cluster_2: Macello_3, Macello_5, Macello_7, Macello_8, Macello_16, Macello_18, Macello_19, Macello_20.*
 - Servito dal *Camion_4*

- *Cluster_3: Macello_2, Macello_4, Macello_10, Macello_14.*
 - Servito dal *Camion_3*

I dati dei clienti contenuti nei cluster, ottenuti mediante questa procedura, diventano elementi di input distinti per la fase successiva di inserzione; l'algoritmo euristico di *Cheapest Insertion*, descritto nel paragrafo successivo, definisce il sequenziamento (*route*) dei clienti contenuti all'interno di ciascun cluster, a cui è già stato assegnato il veicolo.

7.5.3.Applicazione algoritmo Cheapest Insertion

Per ciascuno dei tre cluster precedentemente individuati (vedi paragrafo 7.5.2), è stato trovato il tragitto ottimale che soddisfa le richieste dei clienti; le informazioni che descrivono i processi intermedi di elaborazione dati sono riportate in Appendice, sezione A.3.2.1 per il tragitto numero 1, sezione A.3.2.2 per il tragitto numero 2 e sezione A.3.2.3 per il tragitto numero 3.

Particolare attenzione deve essere posta sul percorso servito dal *Camion_4* in quanto, il tragitto numero 2 ad esso abbinato, non comprende tutti i macelli originariamente contenuti nel *Cluster_2*. Questa situazione può essere compresa analizzando i dati di elaborazione riportati in Appendice, sezione A.3.2.2. In questo caso, infatti, l'algoritmo non è in grado di assegnare il *Macello_3* al tragitto numero 2 perché, qualsiasi sia la sua posizione all'interno dello stesso, non permette il superamento dei vincoli da parte di tutti i clienti coinvolti nel percorso. Come di seguito indicato, il *Macello_3* rimane all'interno dell'array *MacNonAss()* e viene inserito nel cluster successivo (*Cluster_3*) per diventare poi oggetto di analisi dell'algoritmo di *Cheapest Insertion* ad esso applicato.

- *Cluster_3: Macello_2, Macello_3, Macello_4, Macello_10, Macello_14.*
 - Servito dal *Camion_3*

Una volta completata la fase di assegnazione e verifica dei vincoli per ogni cluster, l'algoritmo esegue la *Sub CalcoloCostiTragitto()* per determinare i costi di ciascun percorso individuato; nelle tabelle 7.22, 7.23 e 7.24 sono riportati i costi finali, rispettivamente, del tragitto numero 1, del tragitto numero 2 e del tragitto numero 3.

- **Tragitto numero 1:** Deposito - Macello_1 – Macello_9 – Macello_11 – Macello_15 – Macello_17 – Macello_12 – Macello_13 – Macello_6 – Conceria
 - Servito dal *Camion_1*

TRAGITTO FINALE del *Camion_1*

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_15	2390	120	549	0	669	11	142
Macello_17	1970	161	711	0	872	14	189
Macello_12	1420	190	836	0	1026	18	217
Macello_13	1150	224	956	0	1180	22	247
Macello_6	810	246	1102	0	1348	24	293
Conceria	810	256	1266	0	1522	29	328

TRAGITTO NUMERO 1: costo totale del percorso associato al *Camion_1*

708.2

Tabella 7.22 Costo finale tragitto numero 1

- **Tragitto numero 2:** Deposito - Macello_5 – Macello_16 – Macello_19 – Macello_20 – Macello_18 – Macello_8 – Macello_7 – Conceria
 - Servito dal *Camion_4*

TRAGITTO FINALE del *Camion_4*

Camion_4	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	20	0	0	0	0	0
Macello_5	3310	51	129	61	241	4	42
Macello_16	2960	74	248	61	383	6	82
Macello_19	2450	105	367	61	533	9	127
Macello_20	2170	126	512	61	699	11	157
Macello_18	1790	156	643	61	860	13	191
Macello_8	1400	199	791	61	1051	16	242
Macello_7	980	222	938	61	1221	19	298
Conceria	980	232	1077	61	1370	24	340

TRAGITTO NUMERO 2: costo totale del percorso associato al *Camion_4*

707.95

Tabella 7.23 Costo finale tragitto numero 2

- **Tragitto numero 3:** Deposito - Macello_3 – Macello_2 – Macello_14 – Macello_4 – Macello_10 – Conceria
 - Servito dal *Camion_3*

TRAGITTO FINALE del *Camion_3*

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Macello_14	1170	125	454	3	582	11	143
Macello_4	870	145	566	3	714	13	183
Macello_10	250	192	683	3	878	16	225
Conceria	250	202	815	3	1020	20	269

TRAGITTO NUMERO 3: costo totale del percorso associato al *Camion_3*

568

Tabella 7.24 Costo finale tragitto numero 3

7.5.4. Soluzione finale del problema

La soluzione conclusiva, ossia il costo finale totale del problema oggetto di questo caso studio (20 macelli da servire con 5 mezzi a disposizione), è costituita dalla somma dei costi finali di ogni singolo tragitto individuato dalla procedura. La somma di tutte le variabili *CostoFinaleTragitto*, calcolate per ogni percorso individuato dall'algoritmo, fornisce la soluzione e quindi il costo finale del problema.

In tabella 7.25 è riportato il costo finale del problema (1984.15), dato dalla somma del costo finale tragitto numero 1 (708.2), del costo finale tragitto numero 2 (707.95) e del costo finale tragitto numero 3 (568), a cui si giunge dopo un tempo di elaborazione di circa 30 secondi.

COSTO TOTALE FINALE RELATIVO AL PROBLEMA DI ROUTING

1984.15

Tabella 7.25 Costo finale del problema

CONCLUSIONI

Il caso studio affrontato conferma che le tecniche euristiche di ottimizzazione, proprio per la loro caratteristica di praticità, possono assumere grande rilevanza dal punto di vista operativo e, quindi, trovare largo impiego nelle varie situazioni che si possono presentare nella pratica lavorativa. Ad esempio, le tecniche euristiche di ottimizzazione possono essere utilizzate per risolvere una vasta gamma di problemi di logistica distributiva, come la movimentazione di pezzi tra reparti produttivi, la raccolta e la distribuzione di materiali, la distribuzione di merci da centri di produzione a centri di smistamento.

Il *Vehicle Routing Problem* è un tema centrale per l'organizzazione della distribuzione di beni e servizi. In letteratura è stato affrontato molteplici volte trovando differenti soluzioni a seconda dei vincoli posti e, quindi delle conseguenti limitazioni. Spesso è stato affrontato con l'impiego di tecniche esatte che, mediante l'impostazione di precisi modelli matematici, hanno permesso il raggiungimento di soluzioni ottime. Tali tecniche presentano però forti limitazioni riguardanti i tempi elevati di elaborazione del software dell'ordine di minuti o addirittura ore, le dimensioni ridotte del problema trattato ed i costi onerosi di licenza degli applicativi necessari per l'implementazione.

Questo elaborato affronta il *Capacitated Vehicle Routing Problem with Time Windows* legato al trasporto di pelli bovine fresche da dei macelli (dove sono state lavorate) ad una conceria. L'obiettivo è quello di trovare una soluzione buona, e quindi, differentemente dagli algoritmi esatti, non una soluzione ottima, ma una che si avvicini a questa e sia ottenuta con brevi tempi di risposta impiegando applicativi di larga diffusione. Con questo scopo, è stato implementato un metodo euristico a due fasi del tipo *Cluster first –Route second*, in cui nella prima parte viene applicata una tecnica euristica di clusterizzazione, al fine di accorpare i macelli in gruppi fattibili e convenienti, mentre nella seconda una tecnica euristica di inserzione (*Cheapest Insertion*), per trovare il sequenziamento ottimale dei macelli all'interno di ogni cluster. Il contesto pratico, dove viene applicato tale metodo, è composto inizialmente da uno o più depositi, da cui possono partire 3 veicoli per soddisfare i fabbisogni e le richieste di 13 macelli; resta inteso che l'intero progetto è stato sviluppato in modo parametrico, quindi adattabile a situazioni differenti tra loro. A tal proposito, per testare le potenzialità dell'algoritmo, il contesto di analisi viene esteso ad una rete di clienti costituita da 20 macelli, in cui possono operare fino a 5 mezzi. Lo sviluppo dell'intera procedura, che permette in brevi tempi di elaborazione (circa 30 secondi) di ottenere

tale soluzione, è stato fatto usando il linguaggio di programmazione *Visual Basic for Applications (VBA)* in ambiente *Microsoft Office* (pacchetto applicativo *Excel*).

Il programma, anche se è stato sviluppato per questo specifico caso studio (la raccolta delle pelli fresche che da dei macelli devono essere portate in conceria nel rispetto dei vincoli temporali imposti e considerando i limiti di capacità dei mezzi a disposizione per il servizio), con qualche correttivo, può essere immediatamente estendibile ad analoghe tipologie di problemi o anche leggermente diverse, ma basate sui medesimi principi. Infatti, con minime modifiche, la procedura descritta in questo elaborato può essere estesa a problemi di distribuzione di determinati prodotti o a problemi produttivi. In quest'ultimo caso, ad esempio, i veicoli possono essere sostituiti dalle macchine operatrici necessarie per eseguire la lavorazione ed i macelli dai lotti produttivi da processare.

In definitiva le tecniche euristiche impiegate sono caratterizzate da un tempo di elaborazione quasi nullo, ossia al variare dei dati di input la variazione dell'output è pressoché immediata, e permettono di ottenere buoni risultati, vicini all'ottimo, senza imporre limitazioni specifiche sul software da utilizzare e sulle dimensioni del problema da affrontare.

BIBLIOGRAFIA

C. Filippi, G. Romanin Jacur, 2012, Fresh bovine skin transportation from slaughters to tannery with special time windows and capacity constraints.

M. Fischetti, 1999, *Lezioni di ricerca operativa*, Padova, Italia: Edizione Libreria Progetto.

G. Freschi, 2013, *Un problema complesso di routing con finestre temporali e capacità limitate*, Tesi di laurea magistrale in Ingegneria Gestionale, Università degli Studi di Padova.

C. Frye, W. S. Freeze, F. K. Buckingham, 2003, *Programmare Microsoft Office Excel 2003 con Visual Basic for applications*, Mondadori Informatica

L. Gelli, 2013, *Tecniche euristiche per un problema di routing con finestre temporali e capacità limitate*, Tesi di laurea magistrale in Ingegneria Gestionale, Università degli Studi di Padova.

G. Romanin Jacur, 2014, *Travelling Salesman Problem*.

M. M. Solomon, 1987, "Algorithms for Vehicle routing and Scheduling Problems with Time Windows Constraints", Operations Research.

D. Vigo, 2003, *Algoritmi euristici: metodi costruttivi*, D.E.I.S. Università di Bologna

V. Zampa, 2014, *Tecniche euristiche cluster per risolvere un problema complesso di routing*, Tesi di laurea magistrale in Ingegneria Gestionale, Università degli Studi di Padova.

APPENDICE

A.1.CODICE VBA USATO NEL PROGETTO

A.1.1.Dichiarazione delle variabili usate nell'applicativo

'Variabili usate nella tabulazione durante l'inserimento dati

Public riga As Integer, colonna As Integer, ultimariga As Integer, ultRigaTrag As Integer

'Variabili di tipo standard usate nella gestione dati all'interno dell'applicativo

Public n_camion As Integer, n_macelli As Integer, n_cluster As Integer

Public TempoCarico() As Double, CostoFissoCarico() As Double, TempoViaggio() As Double

Public CostoFissoViaggio() As Double, TempoTotale() As Double, TempoAttesa() As Double

Public CostoAttesa() As Double, CostoTotViaggio() As Double, CostoTotCarico() As Double

Public CostoTotale() As Double

Public ValMax As Double, DMax As Double, Delta As Double, minimo As Double

Public CostoFinaleTragitto As Double, CostoAttesaPrimoMac As Double, CostoTempoTragitto As Double

Public CostoFinaleProblema As Double

Public aC As Double, bC As Double, wC As Double, stC As Double

Public indCam As Integer, indMac As Integer, indMacPrec As Integer, indMacSuc As Integer, _

ind_j As Integer, ind_i As Integer

Public cluster_m() As String, cluster_v() As String

Public nelemxcluster() As Integer, indCluster As Integer, indMacInserito As Integer

Public fineinserzione As Boolean

Public i As Integer, j As Integer, k As Integer, cont As Integer, tr As Integer, lm As Integer

Public MacNonAss() As Integer

Public verifica As Boolean, pVinc As Boolean, sVinc As Boolean, tVinc As Boolean

Public pos As Range

'Varibile di tipo Macello Assegnato

Type MacelloAssegnato

id As String

capacitaResidua As Double

tempoViaggioAss As Double

tempoCaricoAss As Double

tempoAttesaAss As Double

tempoFineServ As Double

costoCaricoAss As Double

costoViaggioAss As Double

End Type

Public tragitto() As MacelloAssegnato

'Varibile di tipo Costo Generato

Type costoGenerato

id As String

costo As Double

indMac1 As Integer

indMac2 As Integer

End Type

Public CostiGenerati() As costoGenerato, CostiGeneratiTot() As costoGenerato, _

MacEsaminati() As costoGenerato

'Varibile di tipo Macello

Type macello

id As String

richiesta As Double

tempoMaxScarico As Double

apertura As Double

chiusura As Double

End Type

Public m As macello, ListaMacelli() As macello, L() As macello, Lprimo() As macello, Kn() As macello

'Varibile di tipo Camion

Type camion

id As String

costoUnTempo As Double

costoUscita As Double

capacita As Double

costoUnAttesa As Double

partenzaMin As Double

partenzaMax As Double

End Type

Public c As camion, ListaCamion() As camion, V() As camion

A.1.2.Interfaccia utente

```
Private Sub TextNMacelli_Exit(ByVal Cancel As MSForms.ReturnBoolean)
If IsNumeric(TextNMacelli) = False Then
    MsgBox "ATTENZIONE: inserire solo valori numerici", vbInformation, "VALORE ERRATO"
    Cancel = True 'impedisce l'uscita dalla text box
    TextNMacelli.SelStart = 0 'imposto il valore iniziale da selezionare
    TextNMacelli.SelLength = Len(TextNMacelli) 'seleziona tutto il contenuto digitato dentro la text box
End If
End Sub

Private Sub TextNCamion_Exit(ByVal Cancel As MSForms.ReturnBoolean)
If IsNumeric(TextNCamion) = False Then
    MsgBox "ATTENZIONE: inserire solo valori numerici", vbInformation, "VALORE ERRATO"
    Cancel = True 'impedisce l'uscita dalla text box
    TextNCamion.SelStart = 0 'imposto il valore iniziale da selezionare
    TextNCamion.SelLength = Len(TextNCamion) 'seleziona tutto il contenuto digitato dentro la text box
End If
End Sub

Public Sub CmdConferma_Click()
n_camion = TextNCamion.Text
n_macelli = TextNMacelli.Text
Unload Me
End Sub
```

A.1.3.Predisposizione foglio di lavoro per input dati

```
Public Sub Inizializzazione()
    'Interfaccia utente per inserimento numero camion e numero macelli
    UserForm1.Show
    'Inserimento foglio per input dati
    Sheets.Add
    ActiveSheet.Select
    ActiveSheet.Name = "Dati_Input"
    Sheets("Dati_Input").Move After:=Sheets("INIZIO")
    'Prima tabella inserimento dati
    PrimaTabInsDati
```


End Sub

A.1.4.Gestione tabelle per l'inserimento dati

'Tabella inserimento Dati Macelli

```
Public Sub PrimaTablInsDati()  
    Range(Cells(1, 1), Cells(1, n_macelli + 3)).Interior.ColorIndex = 6 'giallo  
    Range(Cells(1, 1), Cells(1, n_macelli + 3)).Font.Bold = True  
    Range(Cells(1, 1), Cells(1, n_macelli + 3)).VerticalAlignment = xlCenter  
    Cells(1, 1) = "DATI INPUT: inserire i valori riguardanti i MACELLI"  
    Range(Cells(1, 1), Cells(1, n_macelli + 3)).Merge 'unisce celle  
'scritte in verticale nella tabella  
    Cells(3, 2) = "Richiesta [Kg]"  
    Cells(3, 3) = "Tempo Max Scarico [h]"  
    Cells(3, 4) = "Apertura [h]"  
    Cells(3, 5) = "Chiusura [h]"  
    Range(Cells(3, 2), Cells(3, 5)).Orientation = 90  
    Range(Cells(3, 2), Cells(3, 5)).Font.Bold = True  
    Range(Cells(3, 2), Cells(3, 5)).Interior.ColorIndex = 15 'grigio  
    Range(Cells(3, 2), Cells(5 + n_macelli, 5)).VerticalAlignment = xlCenter  
    Range(Cells(3, 2), Cells(5 + n_macelli, 5)).HorizontalAlignment = xlCenter  
    Cells(4, 1) = "Deposito"  
    Cells(4 + n_macelli + 1, 1) = "Conceria"  
    Range(Cells(4, 1), Cells(4 + n_macelli + 1, 1)).Font.Bold = True  
    Range(Cells(4, 1), Cells(4 + n_macelli + 1, 1)).Interior.ColorIndex = 15 'grigio  
    Range(Cells(4, 1), Cells(4 + n_macelli + 1, 1)).VerticalAlignment = xlCenter  
    For riga = 1 To n_macelli  
        Cells(4 + riga, 1) = "Macello_" & riga  
    Next riga  
    Columns("A:A").AutoFit  
    For riga = 0 To n_macelli + 2  
        For colonna = 1 To 5  
            If Cells(riga + 3, colonna) <> "" Or riga <> 0 Then  
                Cells(riga + 3, colonna).Select  
                Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous  
            End If  
        Next colonna  
    Next riga  
End Sub
```

```

        Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
        Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
        Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
    End If

    If (riga = 1) And (colonna > 1) Then
        Cells(riga + 3, colonna).Interior.ColorIndex = 3 'rosso
        Cells(riga + 3, colonna) = 0
    End If

    If (riga = n_macelli + 2) And ((colonna = 2) Or (colonna = 3)) Then
        Cells(riga + 3, colonna).Interior.ColorIndex = 3 'rosso
        Cells(riga + 3, colonna) = 0
    End If

    Next colonna

Next riga

'creazione pulsante di conferma dati
Set pos = ActiveSheet.Range(Cells(n_macelli + 7, n_macelli), Cells(n_macelli + 7, n_macelli + 3))
ActiveSheet.Buttons.Add(pos.Left, pos.Top, pos.Width, pos.Height).Select
Selection.Name = "Pulsante_Macelli"
Selection.OnAction = "ControlloDatiMacelli"
ActiveSheet.Shapes("Pulsante_Macelli").Select
Selection.Characters.Text = "Clicca per CONFERMA DATI"
Rows(n_macelli + 7).AutoFit
'posiziona il cursore sulla prima cella della tabella per favorire inserimento
Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, _
n_macelli + 3)), Scroll:=True
Range("B5").Select
End Sub

```

'Controllo Dati Macelli

```

Public Sub ControlloDatiMacelli()
    verifica = True

    MsgBox "Inizio verifica dati inseriti", vbInformation, "Verifica Dati"

    For i = 0 To n_macelli + 1
        For j = 0 To 4
            If IsNumeric(Cells(i + 4, j + 2)) = False Then

```

```

        MsgBox "ATTENZIONE: la cella (" & i + 4 & ", " & j + 2 & ") contiene un valore NON _
accettabile", vbInformation, "VALORE ERRATO"

        verifica = False

        Cells(i + 4, j + 2).Interior.ColorIndex = 6

    End If

Next j

Next i

If (verifica = False) Then

    MsgBox "Correggere i valori evidenziati", vbInformation, "VALORE ERRATO"

    verifica = True

Else

    'carico i DATI INPUT riguardanti i MACELLI nella matrice ListaMacelli

    ReDim ListaMacelli(n_macelli + 1)

    For i = 0 To n_macelli + 1

        m.id = Worksheets("Dati_Input").Cells(i + 4, 1)

        m.richiesta = Worksheets("Dati_Input").Cells(i + 4, 2)

        m.tempoMaxScarico = Worksheets("Dati_Input").Cells(i + 4, 3)

        m.apertura = Worksheets("Dati_Input").Cells(i + 4, 4)

        m.chiusura = Worksheets("Dati_Input").Cells(i + 4, 5)

        ListaMacelli(i) = m

    Next i

    ultimariga = n_macelli + 8

    'ottimizza visualizzazione tabella (in alto a sinistra del foglio di lavoro)

    Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, _
n_macelli + 3)), Scroll:=True

    SecondaTabInsDati

End If

End Sub

```

'Tabella inserimento Dati Camion

```

Public Sub SecondaTabInsDati()

    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Interior.ColorIndex = 6 'giallo

    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Font.Bold = True

    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).VerticalAlignment = xlCenter

    Cells(ultimariga + 1, 1) = "DATI INPUT: inserire i valori riguardanti i CAMION"

```

```

Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Merge 'unisce celle
Cells(ultimariga + 3, 2) = "Costo Unitario Tempo [€/h]"
Cells(ultimariga + 3, 3) = "Costo Uscita [€]"
Cells(ultimariga + 3, 4) = "Capacità Massima [Kg]"
Cells(ultimariga + 3, 5) = "Costo Unitario Attesa [€/h]"
Cells(ultimariga + 3, 6) = "Tempo Minimo Partenza [h]"
Cells(ultimariga + 3, 7) = "Tempo Massimo Partenza [h]"
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 7)).Orientation = 90
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 7)).Font.Bold = True
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 7)).Interior.ColorIndex = 15 'grigio
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3 + n_camion, 7)).VerticalAlignment = xlCenter
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3 + n_camion, 7)).HorizontalAlignment = xlCenter
Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_camion - 1, 1)).Font.Bold = True
Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_camion - 1, 1)).Interior.ColorIndex = 15 'grigio
Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_camion - 1, 1)).VerticalAlignment = xlCenter
For riga = 1 To n_camion
    Cells(3 + ultimariga + riga, 1) = "Camion_" & riga
Next riga
Columns("A:A").AutoFit
For riga = 0 To n_camion
    For colonna = 1 To 7
        If Cells(ultimariga + riga + 3, colonna) <> "" Or riga <> 0 Then
            Cells(ultimariga + riga + 3, colonna).Select
            Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
            Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
            Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
            Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
        End If
    Next colonna
Next riga
'Creazione pulsante di conferma dati
Set pos = ActiveSheet.Range(Cells(ultimariga + 5 + n_camion, n_macelli), Cells(ultimariga _
+ 5 + n_camion, n_macelli + 3))
ActiveSheet.Buttons.Add(pos.Left, pos.Top, pos.Width, pos.Height).Select

```

```

Selection.Name = "Pulsante_Camion"
Selection.OnAction = "ControlloDatiCamion"
ActiveSheet.Shapes("Pulsante_Camion").Select
Selection.Characters.Text = "Clicca per CONFERMA DATI"
Rows(ultimariga + 5 + n_camion).AutoFit
'posiziona il cursore sulla prima cella della tabella per favorire inserimento
Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
+ 3)), Scroll:=True
Cells(ultimariga + 4, 2).Select

```

End Sub

'Controllo Dati Camion

```

Public Sub ControlloDatiCamion()
    verifica = True
    MsgBox "Inizio verifica dati inseriti", vbInformation, "Verifica Dati"
    For i = 0 To n_camion - 1
        For j = 0 To 5
            If IsNumeric(Cells(ultimariga + i + 4, j + 2)) = False Then
                MsgBox "ATTENZIONE: la cella (" & i + 4 + ultimariga & ", " & j + 2 & ") contiene un valore _
                NON accettabile", vbInformation, "VALORE ERRATO"
                verifica = False
                Cells(ultimariga + i + 4, j + 2).Select
                Cells(ultimariga + i + 4, j + 2).Interior.ColorIndex = 6
            End If
        Next j
    Next i
    If (verifica = False) Then
        MsgBox "Correggere i valori evidenziati", vbInformation, "VALORE ERRATO"
        verifica = True
    Else
        'carico i DATI INPUT riguardanti i CAMION nella matrice ListaCamion
        ReDim ListaCamion(n_camion - 1)
        For i = 0 To n_camion - 1
            c.id = Worksheets("Dati_Input").Cells(ultimariga + i + 4, 1)
            c.costounTempo = Worksheets("Dati_Input").Cells(ultimariga + i + 4, 2)
        Next i
    End If
End Sub

```

```

c.costoSuscita = Worksheets("Dati_Input").Cells(ultimariga + i + 4, 3)
c.capacita = Worksheets("Dati_Input").Cells(ultimariga + i + 4, 4)
c.costoUnAttesa = Worksheets("Dati_Input").Cells(ultimariga + i + 4, 5)
c.partenzaMin = Worksheets("Dati_Input").Cells(ultimariga + i + 4, 6)
c.partenzaMax = Worksheets("Dati_Input").Cells(ultimariga + i + 4, 7)

ListaCamion(i) = c

Next i

ultimariga = ultimariga + 6 + n_camion

Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
+ 3)), Scroll:=True

TerzaTabInsDati

End If

End Sub

```

'Tabella inserimento Tempi di Carico/Scarico

```

Public Sub TerzaTabInsDati()

Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Interior.ColorIndex = 6 'giallo
Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Font.Bold = True
Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).VerticalAlignment = xlCenter
Cells(ultimariga + 1, 1) = "DATI INPUT: inserire i valori riguardanti i TEMPI di Carico / Scarico"
Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Merge 'unisce celle

For colonna = 1 To n_camion

Cells(ultimariga + 3, colonna + 1) = "Camion_" & colonna

Next colonna

Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 1 + n_camion)).Orientation = 90
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 1 + n_camion)).Font.Bold = True
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 1 + n_camion)).Interior.ColorIndex = 15 'grigio
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3 + n_macelli + 2, _
1 + n_camion)).VerticalAlignment = xlCenter
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3 + n_macelli + 2, _
1 + n_camion)).HorizontalAlignment = xlCenter

Cells(ultimariga + 4, 1) = "Deposito"

Cells(ultimariga + 4 + n_macelli + 1, 1) = "Conceria"

Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_macelli + 1, 1)).Font.Bold = True
Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_macelli + 1, 1)).Interior.ColorIndex = 15 'grigio

```

```

Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_macelli + 1, 1)).VerticalAlignment = xlCenter
For riga = 1 To n_macelli
    Cells(ultimariga + 4 + riga, 1) = "Macello_" & riga
Next riga
Columns("A:A").AutoFit
For riga = 0 To n_macelli + 2
    For colonna = 1 To n_camion + 1
        If Cells(ultimariga + riga + 3, colonna) <> "" Or riga <> 0 Then
            Cells(ultimariga + riga + 3, colonna).Select
            Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
            Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
            Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
            Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
        End If
        If (riga = 1) And (colonna > 1) Then
            Cells(ultimariga + riga + 3, colonna).Interior.ColorIndex = 3 'rosso
            Cells(ultimariga + riga + 3, colonna) = 0
        End If
    Next colonna
Next riga
'creazione pulsante di conferma dati
Set pos = ActiveSheet.Range(Cells(ultimariga + 7 + n_macelli, n_macelli), Cells(ultimariga _
+ 7 + n_macelli, n_macelli + 3))
ActiveSheet.Buttons.Add(pos.Left, pos.Top, pos.Width, pos.Height).Select
Selection.Name = "Pulsante_TempiCaricoScarico"
Selection.OnAction = "ControlloTempiCaricoScarico"
ActiveSheet.Shapes("Pulsante_TempiCaricoScarico").Select
Selection.Characters.Text = "Clicca per CONFERMA DATI"
Rows(ultimariga + 7 + n_macelli).AutoFit
'posiziona il cursore sulla prima cella della tabella per favorire inserimento
Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
+ 3)), Scroll:=True
Cells(ultimariga + 5, 2).Select
End Sub

```

'Controllo Tempi di Carico/Scarico

```
Public Sub ControlloTempiCaricoScarico()

    verifica = True

    MsgBox "Inizio verifica dati inseriti", vbInformation, "Verifica Dati"

    For i = 0 To n_macelli + 1
        For j = 0 To n_camion - 1
            If IsNumeric(Cells(ultimariga + i + 4, j + 2)) = False Then
                MsgBox "ATTENZIONE: la cella (" & ultimariga + i + 4 & ", " & j + 2 & ") contiene un valore _
                NON accettabile", vbInformation, "VALORE ERRATO"
                verifica = False
                Cells(ultimariga + i + 4, j + 2).Select
                Cells(ultimariga + i + 4, j + 2).Interior.ColorIndex = 6
            End If
        Next j
    Next i

    If (verifica = False) Then
        MsgBox "Correggere i valori evidenziati", vbInformation, "VALORE ERRATO"
        verifica = True
    Else
        ' carico i DATI INPUT: TEMPO CARICO / SCARICO nella matrice TempoCarico
        ReDim TempoCarico(n_macelli + 1, n_camion - 1)
        For i = 0 To n_macelli + 1
            For j = 0 To n_camion - 1
                TempoCarico(i, j) = Worksheets("Dati_Input").Cells(ultimariga + i + 4, j + 2)
            Next j
        Next i
        ultimariga = ultimariga + n_macelli + 8
        Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
        + 3)), Scroll:=True
        QuartaTabInsDati
    End If
End Sub
```

'Tabella inserimento Costi di Carico/Scarico

```
Public Sub QuartaTabInsDati()
```



```

Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Interior.ColorIndex = 6 'giallo
Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Font.Bold = True
Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).VerticalAlignment = xlCenter
Cells(ultimariga + 1, 1) = "DATI INPUT: inserire i valori riguardanti i COSTI di Carico / Scarico"
Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Merge 'unisce celle
For colonna = 1 To n_camion
    Cells(ultimariga + 3, colonna + 1) = "Camion_" & colonna
Next colonna
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 1 + n_camion)).Orientation = 90
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 1 + n_camion)).Font.Bold = True
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3, 1 + n_camion)).Interior.ColorIndex = 15 'grigio
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3 + n_macelli + 2, 1 + _
n_camion)).VerticalAlignment = xlCenter
Range(Cells(ultimariga + 3, 2), Cells(ultimariga + 3 + n_macelli + 2, 1 + _
n_camion)).HorizontalAlignment = xlCenter
Cells(ultimariga + 4, 1) = "Deposito"
Cells(ultimariga + 4 + n_macelli + 1, 1) = "Conceria"
Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_macelli + 1, 1)).Font.Bold = True
Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_macelli + 1, 1)).Interior.ColorIndex = 15 'grigio
Range(Cells(ultimariga + 4, 1), Cells(ultimariga + 4 + n_macelli + 1, 1)).VerticalAlignment = xlCenter
For riga = 1 To n_macelli
    Cells(ultimariga + 4 + riga, 1) = "Macello_" & riga
Next riga
Columns("A:A").AutoFit
For riga = 0 To n_macelli + 2
    For colonna = 1 To n_camion + 1
        If Cells(ultimariga + riga + 3, colonna) <> "" Or riga <> 0 Then
            Cells(ultimariga + riga + 3, colonna).Select
            Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
            Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
            Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
            Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
        End If
        If (riga = 1) And (colonna > 1) Then

```

```

        Cells(ultimariga + riga + 3, colonna).Interior.ColorIndex = 3 'rosso
        Cells(ultimariga + riga + 3, colonna) = 0
    End If
Next colonna
Next riga
'creazione pulsante di conferma dati
Set pos = ActiveSheet.Range(Cells(ultimariga + 7 + n_macelli, n_macelli), Cells(ultimariga + 7 _
+ n_macelli, n_macelli + 3))
ActiveSheet.Buttons.Add(pos.Left, pos.Top, pos.Width, pos.Height).Select
Selection.Name = "Pulsante_CostiCaricoScarico"
Selection.OnAction = "ControlloCostiCaricoScarico"
ActiveSheet.Shapes("Pulsante_CostiCaricoScarico").Select
Selection.Characters.Text = "Clicca per CONFERMA DATI"
Rows(ultimariga + 7 + n_macelli).AutoFit
'posiziona il cursore sulla prima cella della tabella per favorire inserimento
Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
+ 3)), Scroll:=True
Cells(ultimariga + 5, 2).Select
End Sub
'Controllo Costi di Carico/Scarico
Public Sub ControlloCostiCaricoScarico()
    verifica = True
    MsgBox "Inizio verifica dati inseriti", vbInformation, "Verifica Dati"
    For i = 0 To n_macelli + 1
        For j = 0 To n_camion - 1
            If IsNumeric(Cells(ultimariga + i + 4, j + 2)) = False Then
                MsgBox "ATTENZIONE: la cella (" & ultimariga + i + 4 & ", " & j + 2 & ") contiene un valore _
                NON accettabile", vbInformation, "VALORE ERRATO"
                verifica = False
                Cells(ultimariga + i + 4, j + 2).Select
                Cells(ultimariga + i + 4, j + 2).Interior.ColorIndex = 6
            End If
        Next j
    Next i
Next i

```

```

If (verifica = False) Then
    MsgBox "Correggere i valori evidenziati", vbInformation, "VALORE ERRATO"
    verifica = True
Else
    'carico i DATI INPUT: COSTO CARICO / SCARICO nella matrice CostoFissoCarico
    ReDim CostoFissoCarico(n_macelli + 1, n_camion - 1)
    For i = 0 To n_macelli + 1
        For j = 0 To n_camion - 1
            CostoFissoCarico(i, j) = Worksheets("Dati_Input").Cells(ultimariga + i + 4, j + 2)
        Next j
    Next i
    ultimariga = ultimariga + n_macelli + 8
    Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
    + 3)), Scroll:=True
    QuintaTabInsDati
End If
End Sub

```

'Tabella inserimento Tempi di Viaggio per camion (deposito-macello-conceria)

```

Public Sub QuintaTabInsDati()
    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Interior.ColorIndex = 6 'giallo
    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Font.Bold = True
    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).VerticalAlignment = xlCenter
    Cells(ultimariga + 1, 1) = "DATI INPUT: inserire i valori riguardanti i TEMPI di VIAGGIO per _
    ogni camion"
    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Merge 'unisce celle
    For cont = 1 To n_camion
        Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2) = "Deposito"
        Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 3 + n_macelli) = "Conceria"
        For colonna = 1 To n_macelli
            Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), colonna + 2) = "Macello_" & colonna
        Next colonna
        Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + ((4 + _
        n_macelli) * (cont - 1)), 3 + n_macelli)).Orientation = 90
        Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + ((4 + _

```

```

n_macelli) * (cont - 1)), 3 + n_macelli).Font.Bold = True
Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + ((4 + _
n_macelli) * (cont - 1)), 3 + n_macelli)).Interior.ColorIndex = 15 'grigio
Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + ((4 + _
n_macelli) * (cont - 1)) + n_macelli + 2, 3 + n_macelli)).VerticalAlignment = xlCenter
Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + ((4 + _
n_macelli) * (cont - 1)) + n_macelli + 2, 3 + n_macelli)).HorizontalAlignment = xlCenter
Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)), 1) = "Deposito"
Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) + n_macelli + 1, 1) = "Conceria"
Range(Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)), 1), Cells(ultimariga + _
(4 * cont) + (n_macelli * (cont - 1)) + n_macelli + 1, 1)).Font.Bold = True
Range(Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)), 1), Cells(ultimariga + _
(4 * cont) + (n_macelli * (cont - 1)) + n_macelli + 1, 1)).Interior.ColorIndex = 15 'grigio
Range(Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)), 1), Cells(ultimariga + _
(4 * cont) + (n_macelli * (cont - 1)) + n_macelli + 1, 1)).VerticalAlignment = xlCenter
For riga = 1 To n_macelli
    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) + riga, 1) = "Macello_" & riga
Next riga
For riga = 0 To n_macelli + 2
    For colonna = 1 To n_macelli + 3
        If Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)) + riga, colonna) <> "" Or riga <> 0 Then
            Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)) + riga, colonna).Select
            Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
            Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
            Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
            Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
        End If
        If ((riga = n_macelli + 2) And (colonna > 1)) Or ((colonna = 2) And (riga > 0)) Or ((riga = 1) _
And (colonna = n_macelli + 3)) Or ((riga = colonna - 1) And (riga > 0)) Then
            Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)) + _
            riga, colonna).Interior.ColorIndex = 3 'rosso
            Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)) + riga, colonna) = 0
        End If
    Next colonna

```

Next riga

```
Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1) = "CAMION_" & cont
```

```
Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).Font.Bold = True
```

```
Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).Font.Size = 12
```

```
Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).Interior.ColorIndex = 6 'giallo
```

```
Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).VerticalAlignment = xlCenter
```

```
Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).HorizontalAlignment = xlCenter
```

Next cont

```
Columns("A:A").AutoFit
```

'creazione pulsante di conferma dati

```
Set pos = ActiveSheet.Range(Cells(ultimariga + 3 + ((4 + n_macelli) * n_camion), _  
n_macelli), Cells(ultimariga + 3 + ((4 + n_macelli) * n_camion), n_macelli + 3))
```

```
ActiveSheet.Buttons.Add(pos.Left, pos.Top, pos.Width, pos.Height).Select
```

```
Selection.Name = "Pulsante_TempiViaggio"
```

```
Selection.OnAction = "ControlloTempiViaggio"
```

```
ActiveSheet.Shapes("Pulsante_TempiViaggio").Select
```

```
Selection.Characters.Text = "Clicca per CONFERMA DATI"
```

```
Rows(ultimariga + 3 + ((4 + n_macelli) * n_camion)).AutoFit
```

'posiziona il cursore sulla prima cella della tabella per favorire inserimento

```
Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, _  
n_macelli + 3)), Scroll:=True
```

```
Cells(ultimariga + 4, 3).Select
```

End Sub

'Controllo Tempi di Viaggio per camion (deposito-macello-conceria)

```
Public Sub ControlloTempiViaggio()
```

```
verifica = True
```

```
MsgBox "Inizio verifica dati inseriti", vbInformation, "Verifica Dati"
```

```
For cont = 1 To n_camion
```

```
For i = 0 To n_macelli + 1
```

```
For j = 0 To n_macelli + 1
```

```
If IsNumeric(Cells(i + ultimariga + (4 * cont) + (n_macelli * (cont - 1)), j + 2)) = False Then
```

```
MsgBox "ATTENZIONE: la cella (" & i + ultimariga + (4 * cont) + (n_macelli * (cont - 1)) _
```

```
& ", " & j + 2 & ") contiene un valore NON accettabile", vbInformation, "VALORE ERRATO"
```

```
verifica = False
```

```

Cells(i + ultimariga + (4 * cont) + (n_macelli * (cont - 1)), j + 2).Select
Cells(i + ultimariga + (4 * cont) + (n_macelli * (cont - 1)), j + 2).Interior.ColorIndex = 6
End If
Next j
Next i
Next cont
If (verifica = False) Then
    MsgBox "Correggere i valori evidenziati", vbInformation, "VALORE ERRATO"
    verifica = True
Else
    'carico i DATI INPUT: valori riguardanti i TEMPI di VIAGGIO per ogni camion nella _
    matrice TempoViaggio
    ReDim TempoViaggio(n_camion - 1, n_macelli + 1, n_macelli + 1)
    For cont = 1 To n_camion
        For i = 0 To n_macelli + 1
            For j = 0 To n_macelli + 1
                TempoViaggio(cont - 1, i, j) = Worksheets("Dati_Input").Cells(i + ultimariga + _
                (4 * cont) + (n_macelli * (cont - 1)), j + 2)
            Next j
        Next i
    Next cont
    ultimariga = ultimariga + 4 + ((4 + n_macelli) * n_camion)
    Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
    + 3)), Scroll:=True
    SestaTabInsDati
End If
End Sub

```

'Tabella inserimento Costi di Viaggio per camion (deposito-macello-conceria)

```

Public Sub SestaTabInsDati()
    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Interior.ColorIndex = 6 'giallo
    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Font.Bold = True
    Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).VerticalAlignment = xlCenter
    Cells(ultimariga + 1, 1) = "DATI INPUT: inserire i valori riguardanti i COSTI di VIAGGIO _
    per ogni camion"

```

```

Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli + 3)).Merge 'unisce celle
For cont = 1 To n_camion
    Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2) = "Deposito"
    Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 3 + n_macelli) = "Conceria"
    For colonna = 1 To n_macelli
        Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), colonna + 2) = "Macello_" & colonna
    Next colonna
    Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + _
    ((4 + n_macelli) * (cont - 1)), 3 + n_macelli)).Orientation = 90
    Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + _
    ((4 + n_macelli) * (cont - 1)), 3 + n_macelli)).Font.Bold = True
    Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + _
    ((4 + n_macelli) * (cont - 1)), 3 + n_macelli)).Interior.ColorIndex = 15 'grigio
    Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + _
    ((4 + n_macelli) * (cont - 1)) + n_macelli + 2, 3 + n_macelli)).VerticalAlignment = xlCenter
    Range(Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)), 2), Cells(ultimariga + 3 + _
    ((4 + n_macelli) * (cont - 1)) + n_macelli + 2, 3 + n_macelli)).HorizontalAlignment = xlCenter
    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)), 1) = "Deposito"
    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) + n_macelli + 1, 1) = "Conceria"
    Range(Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)), 1), Cells(ultimariga + _
    (4 * cont) + (n_macelli * (cont - 1)) + n_macelli + 1, 1)).Font.Bold = True
    Range(Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)), 1), Cells(ultimariga + _
    (4 * cont) + (n_macelli * (cont - 1)) + n_macelli + 1, 1)).Interior.ColorIndex = 15 'grigio
    Range(Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)), 1), Cells(ultimariga + _
    (4 * cont) + (n_macelli * (cont - 1)) + n_macelli + 1, 1)).VerticalAlignment = xlCenter
    For riga = 1 To n_macelli
        Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) + riga, 1) = "Macello_" & riga
    Next riga
    For riga = 0 To n_macelli + 2
        For colonna = 1 To n_macelli + 3
            If Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)) + riga, colonna) <> "" Or riga <> 0 Then
                Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)) + riga, colonna).Select
                Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
                Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
            End If
        Next colonna
    Next riga
Next cont

```

```

        Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
        Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
    End If

    If ((riga = n_macelli + 2) And (colonna > 1)) Or ((colonna = 2) And (riga > 0)) Or ((riga = 1) _
    And (colonna = n_macelli + 3)) Or ((riga = colonna - 1) And (riga > 0)) Then
        Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)) + riga, _
        colonna).Interior.ColorIndex = 3 'rosso
        Cells(ultimariga + 3 + ((4 + n_macelli) * (cont - 1)) + riga, colonna) = 0
    End If

    Next colonna

    Next riga

    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1) = "CAMION_" & cont
    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).Font.Bold = True
    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).Font.Size = 12
    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).Interior.ColorIndex = 6 'giallo
    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).VerticalAlignment = xlCenter
    Cells(ultimariga + (4 * cont) + (n_macelli * (cont - 1)) - 1, 1).HorizontalAlignment = xlCenter

    Next cont

    Columns("A:A").AutoFit

    'creazione pulsante di conferma dati
    Set pos = ActiveSheet.Range(Cells(ultimariga + 3 + ((4 + n_macelli) * n_camion), _
    n_macelli), Cells(ultimariga + 3 + ((4 + n_macelli) * n_camion), n_macelli + 3))
    ActiveSheet.Buttons.Add(pos.Left, pos.Top, pos.Width, pos.Height).Select
    Selection.Name = "Pulsante_CostiViaggio"
    Selection.OnAction = "ControlloCostiViaggio"
    ActiveSheet.Shapes("Pulsante_CostiViaggio").Select
    Selection.Characters.Text = "Clicca per CONFERMA DATI"
    Rows(ultimariga + 3 + ((4 + n_macelli) * n_camion)).AutoFit
    'posiziona il cursore sulla prima cella della tabella per favorire inserimento
    Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
    + 3)), Scroll:=True

    Cells(ultimariga + 4, 3).Select
End Sub

```


'Controllo Costi di Viaggio per camion (deposito-macello-conceria)

```
Public Sub ControlloCostiViaggio()
    verifica = True
    MsgBox "Inizio verifica dati inseriti", vbInformation, "Verifica Dati"
    For cont = 1 To n_camion
        For i = 0 To n_macelli + 1
            For j = 0 To n_macelli + 1
                If IsNumeric(Cells(i + ultimariga + (4 * cont) + (n_macelli * (cont - 1)), j + 2)) = False Then
                    MsgBox "ATTENZIONE: la cella (" & i + ultimariga + (4 * cont) + (n_macelli * (cont - 1)) _
                        & ", " & j + 2 & ") contiene un valore NON accettabile", vbInformation, "VALORE ERRATO"
                    verifica = False
                    Cells(i + ultimariga + (4 * cont) + (n_macelli * (cont - 1)), j + 2).Select
                    Cells(i + ultimariga + (4 * cont) + (n_macelli * (cont - 1)), j + 2).Interior.ColorIndex = 6
                End If
            Next j
        Next i
    Next cont
    If (verifica = False) Then
        MsgBox "Correggere i valori evidenziati", vbInformation, "VALORE ERRATO"
        verifica = True
    Else
        'carico i DATI INPUT: valori riguardanti i COSTI di VIAGGIO per ogni camion nella _
        matrice CostoFissoViaggio
        ReDim CostoFissoViaggio(n_camion - 1, n_macelli + 1, n_macelli + 1)
        For cont = 1 To n_camion
            For i = 0 To n_macelli + 1
                For j = 0 To n_macelli + 1
                    CostoFissoViaggio(cont - 1, i, j) = Worksheets("Dati_Input").Cells(i + ultimariga + _
                        (4 * cont) + (n_macelli * (cont - 1)), j + 2)
                Next j
            Next i
        Next cont
        ultimariga = ultimariga + 4 + ((4 + n_macelli) * n_camion)
        Application.Goto Reference:=Range(Cells(ultimariga + 1, 1), Cells(ultimariga + 1, n_macelli _
```

```

+ 3)), Scroll:=True
ListaMacelliOrdinata
End If
End Sub

```

A.1.5.Algoritmo Cluster

'Passo_1a

```

Sub ListaMacelliOrdinata()
'carico i MACELLI nella matrice L per poi ordinarla
ReDim L(n_macelli - 1)
For i = 1 To n_macelli
    L(i - 1) = ListaMacelli(i)
Next i
'ordinamento crescente per tempi (si, fi) delle finestre temporali
For i = 0 To n_macelli - 2
    For j = 0 To n_macelli - i - 2
        If (L(j).apertura > L(j + 1).apertura) Then
            m = L(j + 1)
            L(j + 1) = L(j)
            L(j) = m
        Else
            If (L(j).apertura = L(j + 1).apertura) Then
                If (L(j).chiusura > L(j + 1).chiusura) Then
                    m = L(j + 1)
                    L(j + 1) = L(j)
                    L(j) = m
                End If
            End If
        End If
    Next j
Next i
ListaCamionOrdinata
End Sub

```

'Passo_1b

```
Sub ListaCamionOrdinata()  
    'carico i CAMION nella matrice V per poi ordinarla  
    ReDim V(n_camion - 1)  
    For i = 0 To n_camion - 1  
        V(i) = ListaCamion(i)  
    Next i  
    'ordinamento DECRESCENTE dei valori del rapporto (capacità massima / costo uscita)  
    For i = 0 To n_camion - 2  
        For j = 0 To n_camion - i - 2  
            If ((V(j).capacita / V(j).costoUscita) < (V(j + 1).capacita / V(j + 1).costoUscita)) Then  
                c = V(j + 1)  
                V(j + 1) = V(j)  
                V(j) = c  
            End If  
        Next j  
    Next i  
    DistanzaMassima  
End Sub
```

'Passo_1c

```
Sub DistanzaMassima()  
    'definisce la distanza massima consentita tra due nodi all'interno di un cluster  
    ValMax = 0  
    For k = 0 To n_camion - 1  
        For i = 1 To n_macelli  
            For j = 1 To n_macelli  
                If (ValMax < CostoFissoViaggio(k, i, j)) Then  
                    ValMax = CostoFissoViaggio(k, i, j)  
                End If  
            Next j  
        Next i  
    Next k  
    DMax = ValMax  
    TempoAttMaxConsentito
```

```

End Sub
Sub TempoAttMaxConsentito()
    'definisce il tempo di attesa massimo consentito (Delta)
    'calcolo Delta come max(apertura(j) - (Chiusura(i)+Tempo(i->j)) valutato per ogni camion
    'k rappresenta i mezzi, i rappresenta ciò che "chiude"
    'j rappresenta ciò che "apre" e che riceve un servizio da un mezzo k transitato in
    ' un nodo i a monte che "chiude"
    Dim temp As Double
    ValMax = 0
    For j = 1 To n_macelli
        For i = 1 To n_macelli
            If (i <> j) Then
                For k = 0 To n_camion - 1
                    temp = ListaMacelli(j).apertura - (ListaMacelli(i).chiusura + TempoViaggio(k, i, j))
                    If (ValMax < temp) Then
                        ValMax = temp
                    End If
                Next k
            End If
        Next i
    Next j
    Delta = ValMax
    AlgoritmoCluster
End Sub
Sub AlgoritmoCluster()
    Dim uscita_V As Integer, uscita_L As Integer, uscita_Lprimo As Integer, uscita_carico As Integer
    Dim uscita_dji As Integer, uscita_bC As Integer, uscita_aj As Integer
    Dim contMac As Integer
    Dim dji As Double, Max_bCbj As Double
    indCluster = 0
    contMac = 0
    uscita_dji = 0
    uscita_V = 0
    uscita_L = 0

```

uscita_Lprimo = 0

uscita_carico = 0

uscita_bC = 0

uscita_aj = 0

ReDim cluster_m(n_camion - 1, n_macelli - 1)

ReDim cluster_v(n_camion - 1)

ReDim nelemxcluster(n_camion - 1)

Do

Do

'Passo_2

'lista vuota di partenza Kn associata al cluster da creare

ReDim Kn(0)

'mezzo associato al cluster da creare e identificazione indice nella matrice ListaCamion

c = V(0)

For i = 0 To n_camion - 1

 If c.id = ListaCamion(i).id Then

 indCam = i

 Exit For

End If

Next i

'aggiornamento dimensione lista veicoli V

If (UBound(V) - 1 >= 0) Then

 For i = 0 To UBound(V) - 1

 V(i) = V(i + 1)

 Next i

 ReDim Preserve V(UBound(V) - 1)

Else

 uscita_V = 1

End If

'Passo_3a

'Primo macello lista Kn e identificazione indice primo macello nella ListaMacelli

Kn(0) = L(0)

contMac = contMac + 1

```

aC = L(0).apertura
bC = L(0).chiusura
wC = L(0).richiesta
For i = 1 To n_macelli
    If L(0).id = ListaMacelli(i).id Then
        indMac = i
        Exit For
    End If
Next i
stC = TempoCarico(indMac, indCam)

```

'Passo_3b

```

If (UBound(L) - 1 >= 0) Then
    For i = 0 To UBound(L) - 1
        L(i) = L(i + 1)
    Next i
    ReDim Preserve L(UBound(L) - 1)
    ReDim Lprimo(UBound(L))
    For i = 0 To UBound(L)
        Lprimo(i) = L(i)
    Next i
Else
    uscita_L = 1
End If
Do
    Do
        Do

```

'Passo_4

PrimoNodoJdaLprimo:

```

    If (Lprimo(0).richiesta + wC > ListaCamion(indCam).capacita) Then
        If (UBound(Lprimo) - 1 >= 0) Then
            For i = 0 To UBound(Lprimo) - 1
                Lprimo(i) = Lprimo(i + 1)
            Next i
            ReDim Preserve Lprimo(UBound(Lprimo) - 1)

```

```

Else
    uscita_Lprimo = 1
End If

Else
    uscita_carico = 1
    For i = 1 To n_macelli
        If Lprimo(0).id = ListaMacelli(i).id Then
            indMac = i
            Exit For
        End If
    Next i
End If

Loop Until ((uscita_carico = 1) Or (uscita_Lprimo = 1))

If (uscita_carico = 1) Then
    'Passo_5a
    ind_j = indMac
    dji = DMax
    For i = 0 To UBound(Kn)
        For j = 1 To n_macelli
            If (Kn(i).id = ListaMacelli(j).id) Then
                If (CostoFissoViaggio(indCam, ind_j, j) < dji) Then
                    dji = CostoFissoViaggio(indCam, ind_j, j)
                    ind_i = j
                End If
            End If
        Exit For
        End If
    Next j
Next i

'Passo_5b
If (dji <= DMax) Then
    uscita_dji = 1
Else
    If (UBound(Lprimo) - 1 >= 0) Then
        For i = 0 To UBound(Lprimo) - 1

```

```

        Lprimo(i) = Lprimo(i + 1)
    Next i

    ReDim Preserve Lprimo(UBound(Lprimo) - 1)

    'inizializzo a zero i vincoli precedenti
    uscita_carico = 0

    Else
        uscita_Lprimo = 1
    End If

End If

End If

End If

Loop Until ((uscita_dji = 1) Or (uscita_Lprimo = 1))

'Passo_6

If (uscita_dji = 1) Then
    If (bC > ListaMacelli(ind_j).chiusura) Then
        Max_bCbj = bC
    Else
        Max_bCbj = ListaMacelli(ind_j).chiusura
    End If

    If (aC + stC + TempoViaggio(indCam, ind_i, ind_j) <= Max_bCbj) Then
        uscita_bC = 1
    Else
        If (UBound(Lprimo) - 1 >= 0) Then
            For i = 0 To UBound(Lprimo) - 1
                Lprimo(i) = Lprimo(i + 1)
            Next i

            ReDim Preserve Lprimo(UBound(Lprimo) - 1)

            'inizializzo a zero i vincoli precedenti
            uscita_carico = 0

            uscita_dji = 0

        Else
            uscita_Lprimo = 1
        End If
    End If

End If

End If

```



```

Loop Until ((uscita_bC = 1) Or (uscita_Lprimo = 1))
If (uscita_bC = 1) Then
    'Passo_7
    If (aC + stC + TempoViaggio(indCam, ind_i, ind_j) + Delta >= ListaMacelli(ind_j).apertura) Then
        uscita_aj = 1
    Else
        For i = 0 To contMac - 1
            cluster_m(indCluster, i) = Kn(i).id
        Next i
        nelemxcluster(indCluster) = contMac
        cluster_v(indCluster) = c.id
        indCluster = indCluster + 1
        contMac = 0
        uscita_carico = 0
        uscita_dji = 0
        uscita_bC = 0
        uscita_aj = 0
        uscita_Lprimo = 0
    End If
End If
Loop Until ((uscita_aj = 1) Or (uscita_Lprimo = 1))
If (uscita_aj = 1) Then
    'Passo_8a
    ReDim Preserve Kn(contMac)
    Kn(contMac) = ListaMacelli(ind_j)
    contMac = contMac + 1
    wC = wC + ListaMacelli(ind_j).richiesta
    If (stC + TempoViaggio(indCam, ind_i, ind_j) > _
        ListaMacelli(ind_j).apertura - ListaMacelli(ind_i).apertura) Then
        stC = stC + TempoViaggio(indCam, ind_i, ind_j) + TempoCarico(ind_j, indCam)
    Else
        stC = ListaMacelli(ind_j).apertura + TempoCarico(ind_j, indCam) - ListaMacelli(ind_i).apertura
    End If
End If

```

'Passo_8b

```
If (bC > ListaMacelli(ind_j).chiusura) Then
    bC = ListaMacelli(ind_j).chiusura
End If

'Elimino j da Lprimo
If (UBound(Lprimo) - 1 >= 0) Then
    For i = 0 To UBound(Lprimo) - 1
        Lprimo(i) = Lprimo(i + 1)
    Next i
    ReDim Preserve Lprimo(UBound(Lprimo) - 1)
    'inizializzo a zero i vincoli precedenti
    uscita_carico = 0
    uscita_dji = 0
    uscita_aj = 0
    uscita_bC = 0
Else
    uscita_Lprimo = 1
End If

'Elimino j da L
For k = 0 To UBound(L)
    If (L(k).id = ListaMacelli(ind_j).id) Then
        If (UBound(L) - 1 >= 0) Then
            If (k <> UBound(L)) Then
                For i = k To UBound(L) - 1
                    L(i) = L(i + 1)
                Next i
            End If
            ReDim Preserve L(UBound(L) - 1)
            'inizializzo a zero i vincoli precedenti
            uscita_carico = 0
            uscita_dji = 0
            uscita_aj = 0
            uscita_bC = 0
        Else
```

```

        uscita_L = 1
    End If
    Exit For
End If
Next k
End If

'Passo_9
If (uscita_Lprimo = 1) Then
    For i = 0 To contMac - 1
        cluster_m(indCluster, i) = Kn(i).id
    Next i
    nelemxcluster(indCluster) = contMac
    cluster_v(indCluster) = c.id
    indCluster = indCluster + 1
    contMac = 0
    uscita_carico = 0
    uscita_dji = 0
    uscita_bC = 0
    uscita_aj = 0
    uscita_Lprimo = 0
Else
    GoTo PrimoNodoJdaLprimo
End If
Loop Until (uscita_L = 1) Or (uscita_V = 1)
If (uscita_V = 1) And (uscita_L = 0) Then
    MsgBox "ATTENZIONE: i mezzi a disposizione hanno capacità complessiva di carico _
    inferiore alla richiesta", vbInformation, "MEZZI INSUFFICIENTI"
Else
    InizioCheapestInsertion
End If
End Sub

```

A.1.6.Algoritmo Cheapest Insertion

A.1.6.1.Predisposizione variabili

Sub InizioCheapestInsertion()

ReDim TempoAttesa(n_camion - 1, n_macelli + 1, n_macelli + 1)

ReDim TempoTotale(n_camion - 1, n_macelli + 1, n_macelli + 1)

ReDim CostoAttesa(n_camion - 1, n_macelli + 1, n_macelli + 1)

ReDim CostoTotViaggio(n_camion - 1, n_macelli + 1, n_macelli + 1)

ReDim CostoTotCarico(n_macelli + 1, n_camion - 1)

ReDim CostoTotale(n_camion - 1, n_macelli + 1, n_macelli + 1)

For k = 0 To n_camion - 1

For i = 0 To n_macelli + 1

For j = 0 To n_macelli + 1

CostoTotCarico(i, k) = CostoFissoCarico(i, k) + _

TempoCarico(i, k) * ListaCamion(k).costoUnTempo

If ((j = 0) Or (i = j) Or (i = n_macelli + 1)) Then

TempoAttesa(k, i, j) = 0

TempoTotale(k, i, j) = 0

CostoAttesa(k, i, j) = 0

CostoTotViaggio(k, i, j) = 0

CostoTotale(k, i, j) = 0

Else

TempoTotale(k, i, j) = TempoCarico(i, k) + TempoViaggio(k, i, j)

If (TempoTotale(k, i, j) >= ListaMacelli(j).apertura) Then

TempoAttesa(k, i, j) = 0

CostoAttesa(k, i, j) = 0

Else

TempoAttesa(k, i, j) = ListaMacelli(j).apertura - TempoTotale(k, i, j)

CostoAttesa(k, i, j) = (ListaMacelli(j).apertura - _

TempoTotale(k, i, j)) * ListaCamion(k).costoUnTempo

End If

CostoTotViaggio(k, i, j) = CostoFissoViaggio(k, i, j) + _

TempoViaggio(k, i, j) * ListaCamion(k).costoUnTempo

CostoTotale(k, i, j) = ListaCamion(k).costoUscita + CostoTotCarico(i, k) + _

CostoTotViaggio(k, i, j) + CostoAttesa(k, i, j)

End If

```

    Next j
Next i
Next k
'INIZIO algoritmo Cheapest Insertion iterato per ogni cluster individuato
n_cluster = indCluster
fineinserzione = False
cont = 0
CostoFinaleTragitto = 0
CostoFinaleProblema = 0
Do While cont < n_cluster
    'Aggiunge in ultima posizione un foglio associato al tragitto in esame
    Sheets.Add
    ActiveSheet.Select
    ActiveSheet.Name = "Tragitto Num." & cont + 1
    Sheets("Tragitto Num." & cont + 1).Select
    Sheets("Tragitto Num." & cont + 1).Move After:=Sheets(Worksheets.Count)
    'Intestazione nuovo tragitto
    Range(Cells(1, 1), Cells(1, 8)).Interior.ColorIndex = 33 'azzurro
    Range(Cells(1, 1), Cells(1, 8)).Font.Bold = True
    Range(Cells(1, 1), Cells(1, 8)).VerticalAlignment = xlCenter
    Cells(1, 1) = "TRAGITTO NUMERO " & cont + 1 & ": ANALISI COSTRUZIONE DETTAGLIATA"
    Range(Cells(1, 1), Cells(1, 8)).Merge 'unisce celle
    ultRigaTrag = 3 'fine parte grafica
'Inizio algoritmo di calcolo
    PrimoMacello (cont)
    CalcoloCostiTragitto
    CostoFinaleProblema = CostoFinaleProblema + CostoFinaleTragitto
'Stampa tragitto finale
    TabellaTragittoFinale
    cont = cont + 1
Loop
    Sheets("Soluzione_Finale").Select
    Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Font.Bold = True
    Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).VerticalAlignment = xlCenter

```

```

Cells(ultimariga, 1) = "COSTO TOTALE FINALE RELATIVO AL PROBLEMA DI ROUTING"
Range(Cells(ultimariga, 1), Cells(ultimariga, 7)).Merge 'unisce celle
Range(Cells(ultimariga, 1), Cells(ultimariga, 7)).HorizontalAlignment = xlRight
Cells(ultimariga, 8).Select
Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
Cells(ultimariga, 8) = CostoFinaleProblema
Cells(ultimariga, 8).Interior.ColorIndex = 46 'arancione
MsgBox "Programma Terminato", vbInformation, "FINE"
End Sub

```

A. 1.6.2. Inizializzazione

```

Sub PrimoMacello(i_Cluster As Integer)
    indCluster = i_Cluster

    'Cheapest Insertion FASE 1: INIZIALIZZAZIONE

    ReDim tragitto(2)
    tragitto(0).id = ListaMacelli(0).id
    tragitto(2).id = ListaMacelli(n_macelli + 1).id
    'trovo l'indice del camion associato al cluster in esame
    For i = 0 To n_camion - 1
        If ListaCamion(i).id = cluster_v(i_Cluster) Then
            indCam = i
        Exit For
    End If
    Next i

    'Carico nel vettore MacNonAss gli indici dei macelli del cluster in esame
    ReDim MacNonAss(nelemxcluster(i_Cluster) - 1)
    For i = 0 To nelemxcluster(i_Cluster) - 1
        MacNonAss(i) = indice_macello(cluster_m(i_Cluster, i))
    Next i

    'capacità residua: il primo nodo del tragitto è il deposito e la sua capacità residua
    'è la capacità del mezzo
    tragitto(0).capacitaResidua = ListaCamion(indCam).capacita

```

'tempo di carico assegnato del deposito è pari al tempo partenza minima del camion

tragitto(0).tempoCaricoAss = ListaCamion(indCam).partenzaMin

'costo di carico assegnato del deposito è zero

tragitto(0).costoCaricoAss = 0

'tempo di viaggio assegnato del deposito è zero

tragitto(0).tempoViaggioAss = 0

'costo di viaggio assegnato del deposito è zero

tragitto(0).costoViaggioAss = 0

'tempo di attesa assegnato del deposito è zero

tragitto(0).tempoAttesaAss = 0

'tempo di fine servizio del deposito è zero

tragitto(0).tempoFineServ = 0

'1°macello (tra deposito e conceria) quello che ha il CostoTotale minimo dal deposito

minimo = CostoTotale(indCam, 0, MacNonAss(0))

indMac = MacNonAss(0)

j = 0

If (UBound(MacNonAss) >= 1) Then

For i = 1 To UBound(MacNonAss)

If (CostoTotale(indCam, 0, MacNonAss(i)) < minimo) Then

minimo = CostoTotale(indCam, 0, MacNonAss(i))

indMac = MacNonAss(i)

j = i

End If

Next i

If (j <> UBound(MacNonAss)) Then

For i = j To UBound(MacNonAss) - 1

MacNonAss(i) = MacNonAss(i + 1)

Next i

End If

ReDim Preserve MacNonAss(UBound(MacNonAss) - 1)

fineinserzione = False

Else

fineinserzione = True

End If

```

tragitto(1).id = ListaMacelli(indMac).id
'Aggiornamento campi del tragitto (Deposito - Primo Macello - Conseria)
For i = 1 To UBound(tragitto)
    indMac = indice_macello(tragitto(i).id)
    indMacPrec = indice_macello(tragitto(i - 1).id)
    'capacità residua
    tragitto(i).capacitaResidua = tragitto(i - 1).capacitaResidua - ListaMacelli(indMac).richiesta
    'tempo di carico assegnato
    tragitto(i).tempoCaricoAss = tragitto(i - 1).tempoCaricoAss + TempoCarico(indMac, indCam)
    'costo di carico assegnato
    tragitto(i).costoCaricoAss = tragitto(i - 1).costoCaricoAss + CostoFissoCarico(indMac, indCam)
    'tempo di viaggio assegnato
    tragitto(i).tempoViaggioAss = tragitto(i - 1).tempoViaggioAss + _
    TempoViaggio(indCam, indMacPrec, indMac)
    'costo di viaggio assegnato
    tragitto(i).costoViaggioAss = tragitto(i - 1).costoViaggioAss + _
    CostoFissoViaggio(indCam, indMacPrec, indMac)
    'tempo di attesa assegnato
    If (ListaMacelli(indMac).apertura < tragitto(i - 1).tempoCaricoAss + _
    tragitto(i).tempoViaggioAss) Then
        tragitto(i).tempoAttesaAss = tragitto(i - 1).tempoAttesaAss
    Else
        tragitto(i).tempoAttesaAss = tragitto(i - 1).tempoAttesaAss + _
        (ListaMacelli(indMac).apertura - (tragitto(i - 1).tempoCaricoAss + tragitto(i).tempoViaggioAss))
    End If
    'tempo di fine servizio
    tragitto(i).tempoFineServ = tragitto(i).tempoCaricoAss + _
    tragitto(i).tempoViaggioAss + tragitto(i).tempoAttesaAss
Next i
indMac = indice_macello(tragitto(1).id)
ind_i = 0
ind_j = n_macelli + 1
TabellaTragitto
Assegnazione

```


End Sub

A. 1.6.3. Assegnazione

Sub Assegnazione()

Dim temp As costoGenerato

Dim primociclo As Boolean, trovato As Boolean, primoMacEsaminato As Boolean, esito As Boolean

'Cheapest Insertion FASE 2 ASSEGNAZIONE

If (fineinserzione = False) Then

NuovaAssegnazione:

ReDim CostiGeneratiTot(UBound(MacNonAss), UBound(tragitto) - 1)

ReDim MacEsaminati(0)

primoMacEsaminato = True

'i cicla tante volte quante sono le coppie di nodi adiacenti all'interno del tragitto

For i = 1 To UBound(tragitto)

 'per ogni coppia di nodi adiacenti creo un array elementi di tipo CostoGenerato

 ReDim CostiGenerati(UBound(MacNonAss))

 indMacPrec = indice_macello(tragitto(i - 1).id)

 indMacSuc = indice_macello(tragitto(i).id)

 For j = 0 To UBound(MacNonAss)

 indMac = MacNonAss(j)

 CostiGenerati(j).id = ListaMacelli(indMac).id

 CostiGenerati(j).indMac1 = indMacPrec

 CostiGenerati(j).indMac2 = indMacSuc

 'calcolo COSTO INSERZIONE

 CostiGenerati(j).costo = _

 CostoTotale(indCam, indMacPrec, indMac) + _

 CostoTotale(indCam, indMac, indMacSuc) - _

 CostoTotale(indCam, indMacPrec, indMacSuc)

 Next j

 'Ordinamento CRESCENTE per il valore Costo del vettore CostiGenerati(j)

 For j = 0 To UBound(CostiGenerati) - 1

 For k = 1 To UBound(CostiGenerati) - j

 If (CostiGenerati(k - 1).costo > CostiGenerati(k).costo) Then

 temp = CostiGenerati(k - 1)

 CostiGenerati(k - 1) = CostiGenerati(k)

```

        CostiGenerati(k) = temp
    End If
Next k
Next j
'COPIARE il vettore nella colonna "i-1" della matrice CostiGeneratiTot
For j = 0 To UBound(CostiGenerati)
    CostiGeneratiTot(j, i - 1) = CostiGenerati(j)
Next j
Next i

```

RicercaCostoInserimentoMinimo:

```

minimo = 0
indMac = 0
indMacPrec = 0
indMacSuc = 0
esito = False

```

'Ricerca nella matrice CostiGeneratiTot del Costo minimo di inserzione

```

primociclo = True
For i = 0 To UBound(MacNonAss)
    For j = 0 To UBound(tragitto) - 1
        'verifica che il macello non sia tra quelli già esaminati
        trovato = False
        If (primoMacEsaminato = False) Then
            For k = 0 To UBound(MacEsaminati)
                If (CostiGeneratiTot(i, j).id = MacEsaminati(k).id) And _
                    (CostiGeneratiTot(i, j).indMac1 = MacEsaminati(k).indMac1) And _
                    (CostiGeneratiTot(i, j).indMac2 = MacEsaminati(k).indMac2) Then
                    trovato = True
                    Exit For
                End If
            Next k
        End If
    Next j
End For
If (trovato = False) Then
    If (primociclo = True) Then
        minimo = CostiGeneratiTot(i, j).costo
    End If
End If

```

```

    indMac = indice_macello(CostiGeneratiTot(i, j).id)
    indMacPrec = CostiGeneratiTot(i, j).indMac1
    indMacSuc = CostiGeneratiTot(i, j).indMac2
    primociclo = False
Else
    If (CostiGeneratiTot(i, j).costo < minimo) Then
        minimo = CostiGeneratiTot(i, j).costo
        indMac = indice_macello(CostiGeneratiTot(i, j).id)
        indMacPrec = CostiGeneratiTot(i, j).indMac1
        indMacSuc = CostiGeneratiTot(i, j).indMac2
    End If
End If
End If
Next j
Next i
'salvo gli indici per le scritte nelle pagine dei tragitti
ind_i = indMacPrec
ind_j = indMacSuc
If (UBound(MacEsaminati) + 1 = (UBound(CostiGeneratiTot, 1) + 1) * _
(UBound(CostiGeneratiTot, 2) + 1)) Then
    'se l'array MacelliEsaminati contiene tutti gli elementi di CostiGeneratiTot significa che non
    'sono possibili ulteriori inserimenti.
    'I macelli contenuti in MacNonAss vengono aggiunti al cluster successivo
    fineinserzione = True
Else
    'Aggiornamento MacEsaminati con i dati del candidato all'inserimento nel tragitto appena trovato
    If (primoMacEsaminato = True) Then
        MacEsaminati(0).id = ListaMacelli(indMac).id
        MacEsaminati(0).costo = minimo
        MacEsaminati(0).indMac1 = indMacPrec
        MacEsaminati(0).indMac2 = indMacSuc
        primoMacEsaminato = False
    Else
        ReDim Preserve MacEsaminati(UBound(MacEsaminati) + 1)

```

```

MacEsaminati(UBound(MacEsaminati)).id = ListaMacelli(indMac).id
MacEsaminati(UBound(MacEsaminati)).costo = minimo
MacEsaminati(UBound(MacEsaminati)).indMac1 = indMacPrec
MacEsaminati(UBound(MacEsaminati)).indMac2 = indMacSuc
End If
ReDim Preserve tragitto(UBound(tragitto) + 1)
'Inserimento macello indMac nel tragitto prima della verifica dei vincoli per conferma
For k = 1 To UBound(tragitto) - 1
    If (tragitto(k - 1).id = ListaMacelli(indMacPrec).id) And _
        (tragitto(k).id = ListaMacelli(indMacSuc).id) Then
        For i = 0 To UBound(tragitto) - k - 1
            tragitto(UBound(tragitto) - i) = tragitto(UBound(tragitto) - (i + 1))
        Next i
        tragitto(k).id = ListaMacelli(indMac).id
        'salvo la posizione del macello inserito nel tragitto (se poi non passa i vincoli devo eliminarlo)
        indMacInserito = k
    Exit For
End If
Next k
'Aggiornamento campi del tragitto prima della verifica vincoli
For i = 1 To UBound(tragitto)
    indMac = indice_macello(tragitto(i).id)
    indMacPrec = indice_macello(tragitto(i - 1).id)
    'capacità residua
    tragitto(i).capacitaResidua = tragitto(i - 1).capacitaResidua - ListaMacelli(indMac).richiesta
    'tempo di carico assegnato
    tragitto(i).tempoCaricoAss = tragitto(i - 1).tempoCaricoAss + TempoCarico(indMac, indCam)
    'costo di carico assegnato
    tragitto(i).costoCaricoAss = tragitto(i - 1).costoCaricoAss + CostoFissoCarico(indMac, indCam)
    'tempo di viaggio assegnato
    tragitto(i).tempoViaggioAss = tragitto(i - 1).tempoViaggioAss + _
        TempoViaggio(indCam, indMacPrec, indMac)
    'costo di viaggio assegnato
    tragitto(i).costoViaggioAss = tragitto(i - 1).costoViaggioAss + _

```

```

CostoFissoViaggio(indCam, indMacPrec, indMac)
'tempo di attesa assegnato
If (ListaMacelli(indMac).apertura < tragitto(i - 1).tempoCaricoAss + _
tragitto(i).tempoViaggioAss) Then
    tragitto(i).tempoAttesaAss = tragitto(i - 1).tempoAttesaAss
Else
    tragitto(i).tempoAttesaAss = tragitto(i - 1).tempoAttesaAss + _
    (ListaMacelli(indMac).apertura - (tragitto(i - 1).tempoCaricoAss + tragitto(i).tempoViaggioAss))
End If
'tempo di fine servizio
tragitto(i).tempoFineServ = tragitto(i).tempoCaricoAss + _
tragitto(i).tempoViaggioAss + tragitto(i).tempoAttesaAss
Next i

```

'Cheapest Insertion FASE 3 VERIFICA VINCOLI

```

esito = verifica_vincoli()
If (esito = False) Then
    StampaRigaNonInserimento
    'togliere il macello dal tragitto perchè non ha superato la verifica vincoli
    For i = indMacInserito To UBound(tragitto) - 1
        tragitto(i) = tragitto(i + 1)
    Next i
    ReDim Preserve tragitto(UBound(tragitto) - 1)

```

GoTo RicercaCostoInserimentoMinimo

```

Else
    indMac = indice_macello(tragitto(indMacInserito).id)
    TabellaTragitto
    'togliere il macello dalla lista MacNonAss perchè è stato assegnato
    If (UBound(MacNonAss) > 0) Then
        For i = 0 To UBound(MacNonAss)
            If (indMac = MacNonAss(i)) Then
                If (i <> UBound(MacNonAss)) Then
                    For k = i To UBound(MacNonAss) - 1
                        MacNonAss(k) = MacNonAss(k + 1)
                    Next k
                End If
            End If
        Next i
    End If

```

```

End If

ReDim Preserve MacNonAss(UBound(MacNonAss) - 1)

Exit For

End If

Next i

Else

    fineinserzione = True

    MacNonAss(0) = 0

End If

If (fineinserzione = False) Then

    'aggiornamento costi d'attesa a seguito dell'inserimento macello in tragitto

    'considero uno alla volta tutti i macelli non ancora assegnati

    For i = 0 To UBound(MacNonAss)

        'considero uno alla volta i macelli assegnati al tragitto

        For k = 1 To UBound(tragitto) - 1

            indMac = indice_macello(tragitto(k).id)

            If (TempoTotale(indCam, indMac, MacNonAss(i)) + tragitto(k).tempoViaggioAss < _
                ListaMacelli(MacNonAss(i)).apertura) Then

                'Calcolo Tempo Attesa

                TempoAttesa(indCam, indMac, MacNonAss(i)) = _
                    ListaMacelli(MacNonAss(i)).apertura - TempoTotale(indCam, indMac, MacNonAss(i))

                'Calcolo Costo Attesa

                CostoAttesa(indCam, indMac, MacNonAss(i)) = _
                    ListaCamion(indCam).costoUnTempo * TempoAttesa(indCam, indMac, MacNonAss(i))

            Else

                TempoAttesa(indCam, indMac, MacNonAss(i)) = 0

                CostoAttesa(indCam, indMac, MacNonAss(i)) = 0

            End If

            'Aggiornamento Costo Totale

            CostoTotale(indCam, indMac, MacNonAss(i)) = _
                ListaCamion(indCam).costoUscita + _
                CostoTotCarico(indMac, indCam) + _
                CostoTotViaggio(indCam, indMac, MacNonAss(i)) + _
                CostoAttesa(indCam, indMac, MacNonAss(i))
        
```

```

        Next k
    Next i
    GoTo NuovaAssegnazione
End If
End If
End If
If (fineinserzione = True) Then
    'Aggiornamento campi del tragitto (le inserzioni sono concluse)
    For i = 1 To UBound(tragitto)
        indMac = indice_macello(tragitto(i).id)
        indMacPrec = indice_macello(tragitto(i - 1).id)
        'capacità residua
        tragitto(i).capacitaResidua = tragitto(i - 1).capacitaResidua - ListaMacelli(indMac).richiesta
        'tempo di carico assegnato
        tragitto(i).tempoCaricoAss = tragitto(i - 1).tempoCaricoAss + TempoCarico(indMac, indCam)
        'costo di carico assegnato
        tragitto(i).costoCaricoAss = tragitto(i - 1).costoCaricoAss + CostoFissoCarico(indMac, indCam)
        'tempo di viaggio assegnato
        tragitto(i).tempoViaggioAss = tragitto(i - 1).tempoViaggioAss + _
        TempoViaggio(indCam, indMacPrec, indMac)
        'costo di viaggio assegnato
        tragitto(i).costoViaggioAss = tragitto(i - 1).costoViaggioAss + _
        CostoFissoViaggio(indCam, indMacPrec, indMac)
        'tempo di attesa assegnato
        If (ListaMacelli(indMac).apertura < tragitto(i - 1).tempoCaricoAss + _
        tragitto(i).tempoViaggioAss) Then
            tragitto(i).tempoAttesaAss = tragitto(i - 1).tempoAttesaAss
        Else
            tragitto(i).tempoAttesaAss = tragitto(i - 1).tempoAttesaAss + _
            (ListaMacelli(indMac).apertura - (tragitto(i - 1).tempoCaricoAss + tragitto(i).tempoViaggioAss))
        End If
        'tempo di fine servizio
        tragitto(i).tempoFineServ = tragitto(i).tempoCaricoAss + _
        tragitto(i).tempoViaggioAss + tragitto(i).tempoAttesaAss
    Next i
End If

```

```

Next i

'Inserzioni possibili concluse, carico i macelli non assegnati nel cluster successivo

If (MacNonAss(0) <> 0) Then
    If (indCluster < n_camion - 1) Then
        For i = 0 To UBound(MacNonAss)
            cluster_m(indCluster + 1, nelemxcluster(indCluster + 1) ) = ListaMacelli(MacNonAss(i)).id
            nelemxcluster(indCluster + 1) = nelemxcluster(indCluster + 1) + 1
            nelemxcluster(indCluster) = nelemxcluster(indCluster) - 1
        Next i
        If (indCluster = n_cluster - 1) Then
            n_cluster = n_cluster + 1
            cluster_v(indCluster + 1) = V(0).id
            If (UBound(V) > 0) Then
                For i = 0 To UBound(V) - 1
                    V(i) = V(i + 1)
                Next i
                ReDim Preserve V(UBound(V) - 1)
            Else
                ReDim V(0)
                V(0).id = "FineMezzi"
            End If
        End If
    Else
        MsgBox "Non tutti i macelli possono essere visitati con i mezzi a disposizione nel _
        rispetto dei vincoli", vbInformation, "ATTENZIONE"
    End If
End If

Else
    MsgBox "Verificare codice (fineinserzione = True)", vbInformation, "DEBUG"
End If

Else
    'Caso in cui nel tragitto è presente solo un macello e ho concluso le inserzioni
End If

'aggiorno cluster_m con i macelli effettivi del tragitto

```



```

For i = 0 To n_macelli - 1
  If (i < UBound(tragitto) - 1) Then
    cluster_m(indCluster, i) = tragitto(i + 1).id
  Else
    cluster_m(indCluster, i) = ""
  End If
Next i
End Sub

```

A.1.6.4. Verifica dei vincoli e visualizzazione dati di elaborazione

'Verifica dei vincoli per confermare l'assegnazione

```

Public Function verifica_vincoli() As Boolean
  Dim q As Integer, im As Integer, totVincoliOK As Integer
  Dim I_Vincolo As Boolean, II_Vincolo As Boolean, III_Vincolo As Boolean
  I_Vincolo = False
  II_Vincolo = False
  III_Vincolo = False
  totVincoliOK = 0
  pVinc = True
  sVinc = True
  tVinc = True
  tr = 0
  lm = 0
  For q = 1 To UBound(tragitto)
    im = indice_macello(tragitto(q).id)
    'Vincolo tempo massimo di scarico pelli fresche
    If (tragitto(q).id = ListaMacelli(n_macelli + 1).id) Then
      'se si tratta della conceria il primo vincolo è true per default
      I_Vincolo = True
    Else
      If (tragitto(q).tempoFineServ <= ListaMacelli(im).tempoMaxScarico) Then
        I_Vincolo = True
      Else
        pVinc = False
        tr = q
      End If
    End If
  Next q
End Function

```

```

        lm = im
    End If
End If

'Vincolo tempo chiusura del macello'
If (tragitto(q).tempoFineServ <= ListaMacelli(im).chiusura) Then
    II_Vincolo = True
Else
    sVinc = False
    tr = q
    lm = im
End If

'Vincolo capacita' residua'
If (tragitto(q).capacitaResidua >= 0) Then
    III_Vincolo = True
Else
    tVinc = False
    tr = q
    lm = im
End If

If ((I_Vincolo = True) And (II_Vincolo = True) And (III_Vincolo = True)) Then
    totVincoliOK = totVincoliOK + 1
Else
    totVincoliOK = 0
    Exit For
End If

I_Vincolo = False
II_Vincolo = False
III_Vincolo = False
Next q

If (totVincoliOK = UBound(tragitto)) Then
    verifica_vincoli = True
Else
    verifica_vincoli = False
End If

```

End Function

Public Function indice_macello(nome As String) As Integer

Dim punt As Integer

For punt = 0 To n_macelli + 1

If (nome = ListaMacelli(punt).id) Then

indice_macello = punt

Exit For

End If

Next punt

End Function

'Visualizzazione informazioni del tragitto in costruzione in caso di assegnazione positiva

Sub TabellaTragitto()

Dim h As Integer

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Color = vbBlue

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter

If (ind_j <> n_macelli + 1) Then

Cells(ultRigaTrag, 1) = "Nel tragitto è stato aggiunto il " & ListaMacelli(indMac).id _

& " tra il " & ListaMacelli(ind_j).id & " e il " & ListaMacelli(ind_j).id

Else

'ind_j è la conceria

Cells(ultRigaTrag, 1) = "Nel tragitto è stato aggiunto il " & ListaMacelli(indMac).id _

& " tra il " & ListaMacelli(ind_j).id & " e la " & ListaMacelli(ind_j).id

End If

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Merge 'unisce celle

ultRigaTrag = ultRigaTrag + 1

'Titoli tabella

Cells(ultRigaTrag, 1).Interior.ColorIndex = 43 'verde

Cells(ultRigaTrag, 1) = ListaCamion(indCam).id

Range(Cells(ultRigaTrag, 2), Cells(ultRigaTrag, 8)).Interior.ColorIndex = 15 'grigio chiaro

Cells(ultRigaTrag, 2) = "CAPACITA"

Cells(ultRigaTrag, 3) = "TEMPO CARICO"

Cells(ultRigaTrag, 4) = "TEMPO VIAGGIO"

Cells(ultRigaTrag, 5) = "TEMPO ATTESA"

Cells(ultRigaTrag, 6) = "TEMPO FINE SERVIZIO"

```

Cells(ultRigaTrag, 7) = "COSTO CARICO"
Cells(ultRigaTrag, 8) = "COSTO VIAGGIO"
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Bold = True
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).HorizontalAlignment = xlCenter
'costruzione griglia
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Select
Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
Selection.Borders(xlInsideVertical).LineStyle = xlContinuous
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).WrapText = True 'testo a capo
Columns(1).ColumnWidth = 15 'larghezza colonne
Columns(2).ColumnWidth = 15
Columns(3).ColumnWidth = 15
Columns(4).ColumnWidth = 15
Columns(5).ColumnWidth = 15
Columns(6).ColumnWidth = 15
Columns(7).ColumnWidth = 15
Columns(8).ColumnWidth = 15
ultRigaTrag = ultRigaTrag + 1
For h = 0 To UBound(tragitto)
    Cells(ultRigaTrag, 1) = tragitto(h).id
    If (tragitto(h).id = ListaMacelli(indMac).id) Then
        Cells(ultRigaTrag, 1).Font.Color = vbBlue
    End If
    Cells(ultRigaTrag, 2) = tragitto(h).capacitaResidua
    Cells(ultRigaTrag, 3) = tragitto(h).tempoCaricoAss
    Cells(ultRigaTrag, 4) = tragitto(h).tempoViaggioAss
    Cells(ultRigaTrag, 5) = tragitto(h).tempoAttesaAss
    Cells(ultRigaTrag, 6) = tragitto(h).tempoFineServ
    Cells(ultRigaTrag, 7) = tragitto(h).costoCaricoAss
    Cells(ultRigaTrag, 8) = tragitto(h).costoViaggioAss

```

```

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Select
Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
Selection.Borders(xlInsideVertical).LineStyle = xlContinuous
Cells(ultRigaTrag, 1).Font.Bold = True
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).HorizontalAlignment = xlCenter
ultRigaTrag = ultRigaTrag + 1

```

Next h

ultRigaTrag = ultRigaTrag + 1

End Sub

'Visualizzazione motivazioni nel caso di assegnazione negativa

Sub StampaRigaNonInserimento()

```

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Color = vbRed

```

```

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter

```

```

If (ind_j <> n_macelli + 1) Then

```

```

    Cells(ultRigaTrag, 1) = "Nel tragitto NON è stato possibile aggiungere il " & _
    tragitto(indMacInserito).id & " tra il " & ListaMacelli(ind_i).id & " e il " & ListaMacelli(ind_j).id

```

```

Else

```

```

    'ind_j è la conceria

```

```

    Cells(ultRigaTrag, 1) = "Nel tragitto NON è stato possibile aggiungere il " & _
    tragitto(indMacInserito).id & " tra il " & ListaMacelli(ind_i).id & " e la " & ListaMacelli(ind_j).id

```

```

End If

```

```

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Merge 'unisce celle

```

```

ultRigaTrag = ultRigaTrag + 1

```

```

'Tempo massimo scarico pelli

```

```

If (pVinc = False) Then

```

```

    Cells(ultRigaTrag, 1) = "Il tempo di fine servizio (" & tragitto(tr).tempoFineServ & ") presso il " & _
    & ListaMacelli(lm).id & " è superiore al tempo massimo di scarico delle pelli (" & _
    & ListaMacelli(lm).tempoMaxScarico & ")"

```

```

    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Color = vbRed

```

```

    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter

```

```

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Merge 'unisce celle
ultRigaTrag = ultRigaTrag + 1
End If
'Tempo fine servizio – chiusura del macello
If (sVinc = False) Then
    If (lm = n_macelli + 1) Then
        ' lm rappresenta la conceria
        Cells(ultRigaTrag, 1) = "Il tempo di fine servizio (" & tragitto(tr).tempoFineServ & ") presso la " _
        & ListaMacelli(lm).id & " è superiore al tempo di chiusura della stessa (" _
        & ListaMacelli(lm).chiusura & ")"
    Else
        Cells(ultRigaTrag, 1) = "Il tempo di fine servizio (" & tragitto(tr).tempoFineServ & ") presso il " _
        & ListaMacelli(lm).id & " è superiore al tempo di chiusura dello stesso (" _
        & ListaMacelli(lm).chiusura & ")"
    End If
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Color = vbRed
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Merge 'unisce celle
    ultRigaTrag = ultRigaTrag + 1
End If
'Capacità resiuva
If (tVinc = False) Then
    If (lm = n_macelli + 1) Then
        ' lm rappresenta la conceria
        Cells(ultRigaTrag, 1) = "La capacità residua del " & ListaCamion(indCam).id & " presso la " _
        & ListaMacelli(lm).id & " è negativa (" & tragitto(q).capacitaResidua & ")"
    Else
        Cells(ultRigaTrag, 1) = "La capacità residua del " & ListaCamion(indCam).id & " presso il " _
        & ListaMacelli(lm).id & " è negativa (" & tragitto(q).capacitaResidua & ")"
    End If
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Color = vbRed
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Merge 'unisce celle
    ultRigaTrag = ultRigaTrag + 1

```

End If

ultRigaTrag = ultRigaTrag + 1

End Sub

A.1.6.5. Costi finali del tragitto

'Calcolo costo finale del tragitto

Sub CalcoloCostiTragitto()

'Calcolo del costo d'attesa per il primo macello visitato

If (tragitto(1).tempoAttesaAss + tragitto(0).tempoCaricoAss < ListaCamion(indCam).partenzaMax) Then

CostoAttesaPrimoMac = tragitto(1).tempoAttesaAss * ListaCamion(indCam).costoUnAttesa

Else

indMac = indice_macello(tragitto(1).id)

CostoAttesaPrimoMac = ListaCamion(indCam).costoUnAttesa * _

(ListaCamion(indCam).partenzaMax - tragitto(0).tempoCaricoAss) + _

ListaCamion(indCam).costoUnTempo * (ListaMacelli(indMac).apertura - _

(ListaCamion(indCam).partenzaMax + tragitto(1).tempoViaggioAss))

End If

'Calcolo del costo del tempo di viaggio totale

If (tragitto(1).tempoAttesaAss > 0) Then

CostoTempoTragitto = (tragitto(UBound(tragitto)).tempoFineServ - tragitto(1).tempoAttesaAss) * _

ListaCamion(indCam).costoUnTempo

Else

CostoTempoTragitto = (tragitto(UBound(tragitto)).tempoFineServ - tragitto(0).tempoCaricoAss) * _

ListaCamion(indCam).costoUnTempo

End If

'Costo finale del tragitto

CostoFinaleTragitto = _

ListaCamion(indCam).costoUscita + _

tragitto(UBound(tragitto)).costoViaggioAss + _

tragitto(UBound(tragitto)).costoCaricoAss + _

CostoAttesaPrimoMac + CostoTempoTragitto

End Sub

'Stampa tabella tragitto finale

Sub TabellaTragittoFinale()

Dim h As Integer

'Stampa tabella tragitto finale nella scheda tragitto num #

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Color = vbBlue

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter

Cells(ultRigaTrag, 1) = "TRAGITTO FINALE del " & ListaCamion(indCam).id

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Bold = True

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Merge 'unisce celle

ultRigaTrag = ultRigaTrag + 1

'Titoli tabella

Cells(ultRigaTrag, 1).Interior.ColorIndex = 43 'verde

Cells(ultRigaTrag, 1) = ListaCamion(indCam).id

Range(Cells(ultRigaTrag, 2), Cells(ultRigaTrag, 8)).Interior.ColorIndex = 15 'grigio chiaro

Cells(ultRigaTrag, 2) = "CAPACITA"

Cells(ultRigaTrag, 3) = "TEMPO CARICO"

Cells(ultRigaTrag, 4) = "TEMPO VIAGGIO"

Cells(ultRigaTrag, 5) = "TEMPO ATTESA"

Cells(ultRigaTrag, 6) = "TEMPO FINE SERVIZIO"

Cells(ultRigaTrag, 7) = "COSTO CARICO"

Cells(ultRigaTrag, 8) = "COSTO VIAGGIO"

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Bold = True

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).HorizontalAlignment = xlCenter

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Select 'griglia

Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous

Selection.Borders(xlEdgeRight).LineStyle = xlContinuous

Selection.Borders(xlEdgeTop).LineStyle = xlContinuous

Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous

Selection.Borders(xlInsideVertical).LineStyle = xlContinuous

Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).WrapText = True 'testo a capo

Columns(1).ColumnWidth = 15 'larghezza colonne

Columns(2).ColumnWidth = 15

Columns(3).ColumnWidth = 15

Columns(4).ColumnWidth = 15

Columns(5).ColumnWidth = 15


```

Columns(6).ColumnWidth = 15
Columns(7).ColumnWidth = 15
Columns(8).ColumnWidth = 15
ultRigaTrag = ultRigaTrag + 1
For h = 0 To UBound(tragitto)
    Cells(ultRigaTrag, 1) = tragitto(h).id
    Cells(ultRigaTrag, 2) = tragitto(h).capacitaResidua
    Cells(ultRigaTrag, 3) = tragitto(h).tempoCaricoAss
    Cells(ultRigaTrag, 4) = tragitto(h).tempoViaggioAss
    Cells(ultRigaTrag, 5) = tragitto(h).tempoAttesaAss
    Cells(ultRigaTrag, 6) = tragitto(h).tempoFineServ
    Cells(ultRigaTrag, 7) = tragitto(h).costoCaricoAss
    Cells(ultRigaTrag, 8) = tragitto(h).costoViaggioAss
    If (h = UBound(tragitto)) Then
        Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Bold = True
    End If
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Select 'griglia
    Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
    Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
    Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
    Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
    Selection.Borders(xlInsideVertical).LineStyle = xlContinuous
    Cells(ultRigaTrag, 1).Font.Bold = True
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter
    Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).HorizontalAlignment = xlCenter
    ultRigaTrag = ultRigaTrag + 1
Next h
ultRigaTrag = ultRigaTrag + 1
'Stampa costo totale tragitto
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).Font.Bold = True
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 8)).VerticalAlignment = xlCenter
Cells(ultRigaTrag, 1) = "TRAGITTO NUMERO " & cont + 1 & ": costo totale del percorso _
associato al " & ListaCamion(indCam).id
Range(Cells(ultRigaTrag, 1), Cells(ultRigaTrag, 7)).Merge 'unisce celle

```

```

Cells(ultRigaTrag, 8).Select
Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
Cells(ultRigaTrag, 8) = CostoFinaleTragitto
Cells(ultRigaTrag, 8).Interior.ColorIndex = 6 'giallo

```

'Stampa tabella tragitto finale nella scheda Soluzione_Finale

```

If (cont = 0) Then
    Sheets.Add
    ActiveSheet.Select
    ActiveSheet.Name = "Soluzione_Finale"
    Sheets("Soluzione_Finale").Select
    Sheets("Soluzione_Finale").Move After:=Sheets("Dati_Input")
    'Intestazione nuovo tragitto
    Range(Cells(1, 1), Cells(1, 8)).Interior.ColorIndex = 33 'azzurro
    Range(Cells(1, 1), Cells(1, 8)).Font.Bold = True
    Range(Cells(1, 1), Cells(1, 8)).VerticalAlignment = xlCenter
    Cells(1, 1) = "SOLUZIONE FINALE DEL PROBLEMA DI ROUTING"
    Range(Cells(1, 1), Cells(1, 8)).Merge 'unisce celle
    ultimariga = 3 'fine parte grafica
Else
    Sheets("Soluzione_Finale").Select
End If
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Font.Color = vbBlue
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).VerticalAlignment = xlCenter
Cells(ultimariga, 1) = "TRAGITTO FINALE del " & ListaCamion(indCam).id
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Font.Bold = True
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Merge 'unisce celle
ultimariga = ultimariga + 1
'Titoli tabella
Cells(ultimariga, 1).Interior.ColorIndex = 43 'verde
Cells(ultimariga, 1) = ListaCamion(indCam).id
Range(Cells(ultimariga, 2), Cells(ultimariga, 8)).Interior.ColorIndex = 15 'grigio chiaro

```

```

Cells(ultimariga, 2) = "CAPACITA"
Cells(ultimariga, 3) = "TEMPO CARICO"
Cells(ultimariga, 4) = "TEMPO VIAGGIO"
Cells(ultimariga, 5) = "TEMPO ATTESA"
Cells(ultimariga, 6) = "TEMPO FINE SERVIZIO"
Cells(ultimariga, 7) = "COSTO CARICO"
Cells(ultimariga, 8) = "COSTO VIAGGIO"
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Font.Bold = True
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).VerticalAlignment = xlCenter
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).HorizontalAlignment = xlCenter
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Select
Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
Selection.Borders(xlInsideVertical).LineStyle = xlContinuous
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).WrapText = True
Columns(1).ColumnWidth = 15
Columns(2).ColumnWidth = 15
Columns(3).ColumnWidth = 15
Columns(4).ColumnWidth = 15
Columns(5).ColumnWidth = 15
Columns(6).ColumnWidth = 15
Columns(7).ColumnWidth = 15
Columns(8).ColumnWidth = 15
ultimariga = ultimariga + 1
For h = 0 To UBound(tragitto)
    Cells(ultimariga, 1) = tragitto(h).id
    Cells(ultimariga, 2) = tragitto(h).capacitaResidua
    Cells(ultimariga, 3) = tragitto(h).tempoCaricoAss
    Cells(ultimariga, 4) = tragitto(h).tempoViaggioAss
    Cells(ultimariga, 5) = tragitto(h).tempoAttesaAss
    Cells(ultimariga, 6) = tragitto(h).tempoFineServ
    Cells(ultimariga, 7) = tragitto(h).costoCaricoAss

```

```

Cells(ultimariga, 8) = tragitto(h).costoViaggioAss
If (h = UBound(tragitto)) Then
    Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Font.Bold = True
End If

Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Select
Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
Selection.Borders(xlInsideVertical).LineStyle = xlContinuous

Cells(ultimariga, 1).Font.Bold = True

Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).VerticalAlignment = xlCenter
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).HorizontalAlignment = xlCenter

ultimariga = ultimariga + 1

Next h

ultimariga = ultimariga + 1

'Stampa costo totale tragitto

Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).Font.Bold = True
Range(Cells(ultimariga, 1), Cells(ultimariga, 8)).VerticalAlignment = xlCenter
Cells(ultimariga, 1) = "TRAGITTO NUMERO " & cont + 1 & ": costo totale del percorso _
associato al " & ListaCamion(indCam).id
Range(Cells(ultimariga, 1), Cells(ultimariga, 7)).Merge 'unisce celle
Range(Cells(ultimariga, 1), Cells(ultimariga, 7)).HorizontalAlignment = xlRight
Cells(ultimariga, 8).Select
Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous

Cells(ultimariga, 8) = CostoFinaleTragitto
Cells(ultimariga, 8).Interior.ColorIndex = 6 'giallo

ultimariga = ultimariga + 2

End Sub

```


A.2.TABELLE DEL PROBLEMA CON 13 MACELLI E 3 CAMION

A.2.1.Dati di Input

A.2.1.1.Tempi di viaggio per ogni camion

CAMION_1	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10	Macello_11	Macello_12	Macello_13	Conceria
Deposito	0	125	137	151	134	129	120	90	112	106	149	120	123	141	0
Macello_1	0	0	145	161	156	122	125	111	137	108	149	166	139	122	160
Macello_2	0	142	0	150	160	167	156	163	123	175	141	111	157	152	150
Macello_3	0	158	149	0	168	168	141	128	181	148	124	154	146	139	143
Macello_4	0	166	167	158	0	171	152	181	114	166	116	139	112	148	130
Macello_5	0	156	181	145	132	0	120	100	161	173	146	182	151	116	165
Macello_6	0	112	107	119	137	113	0	115	101	150	161	143	138	144	164
Macello_7	0	109	111	133	112	100	128	0	98	146	151	105	167	150	141
Macello_8	0	131	140	129	107	119	170	159	0	121	177	133	149	101	126
Macello_9	0	103	177	145	170	138	152	149	119	0	133	126	160	175	156
Macello_10	0	152	138	120	115	143	166	153	174	131	0	168	123	169	149
Macello_11	0	168	113	157	136	184	147	107	129	130	124	0	157	160	170
Macello_12	0	134	160	144	115	148	136	172	151	164	170	154	0	120	107
Macello_13	0	154	137	178	151	118	146	149	103	172	169	158	121	0	139
Conceria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CAMION_2	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10	Macello_11	Macello_12	Macello_13	Conceria
Deposito	0	130	140	157	144	133	121	92	120	109	153	121	129	144	0
Macello_1	0	0	147	166	159	123	127	112	139	111	154	166	142	119	150
Macello_2	0	143	0	154	163	165	156	166	125	177	144	115	158	154	166
Macello_3	0	163	151	0	171	166	144	131	187	151	128	159	148	141	115
Macello_4	0	168	170	165	0	170	154	190	116	170	119	141	114	150	159
Macello_5	0	157	178	147	133	0	120	101	170	175	148	186	155	119	141
Macello_6	0	115	109	117	135	115	0	116	107	153	165	148	141	146	186
Macello_7	0	110	113	140	112	102	132	0	99	149	153	109	170	153	148
Macello_8	0	134	143	132	109	121	170	161	0	123	180	136	151	105	133
Macello_9	0	105	183	147	175	142	155	153	123	0	135	131	164	178	140
Macello_10	0	156	141	124	118	147	171	155	177	134	0	172	126	171	158
Macello_11	0	174	118	161	142	190	151	112	132	134	128	0	160	162	147
Macello_12	0	137	165	147	118	152	138	177	154	167	175	156	0	123	113
Macello_13	0	157	140	184	153	119	150	151	104	175	174	162	123	0	153
Conceria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CAMION_3	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10	Macello_11	Macello_12	Macello_13	Conceria
Deposito	0	132	143	160	148	138	121	96	124	115	157	122	133	146	0
Macello_1	0	0	146	163	157	122	126	111	138	109	151	166	140	121	187
Macello_2	0	142	0	152	161	166	157	164	124	176	142	112	158	153	116
Macello_3	0	160	150	0	169	167	142	129	184	149	125	156	147	140	170
Macello_4	0	167	168	150	0	171	153	184	115	167	117	140	113	149	107
Macello_5	0	156	179	146	133	0	121	101	164	174	147	184	153	117	124
Macello_6	0	113	108	118	136	114	0	115	104	151	163	145	139	145	116
Macello_7	0	110	112	136	113	101	129	0	98	147	152	106	168	151	146
Macello_8	0	132	141	130	108	120	169	160	0	122	178	134	150	102	128
Macello_9	0	104	179	146	172	140	153	151	121	0	134	128	162	176	177
Macello_10	0	154	139	121	116	144	168	154	175	132	0	169	124	170	132
Macello_11	0	170	115	159	139	186	149	109	130	131	126	0	158	161	154
Macello_12	0	135	162	145	116	150	137	174	152	165	172	155	0	121	104
Macello_13	0	155	138	181	152	119	148	150	103	174	171	160	122	0	128
Conceria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.1.2.Costi di viaggio per ogni camion

CAMION_1	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10	Macello_11	Macello_12	Macello_13	Conceria
Deposito	0	25	37	51	34	41	33	28	31	33	29	40	35	32	0
Macello_1	0	0	45	61	56	49	51	46	39	36	48	35	51	46	41
Macello_2	0	42	0	50	60	56	60	51	57	63	47	55	44	51	39
Macello_3	0	58	49	0	68	64	51	60	42	40	31	55	50	67	50
Macello_4	0	66	67	58	0	71	66	72	61	66	43	50	31	44	45
Macello_5	0	45	58	49	58	0	61	59	77	44	53	69	52	36	41
Macello_6	0	59	71	49	61	80	0	67	63	52	75	40	62	45	35
Macello_7	0	75	65	49	51	49	39	0	29	63	42	42	76	38	37
Macello_8	0	53	45	71	47	69	70	54	0	36	64	54	57	38	26
Macello_9	0	35	65	42	67	43	54	61	38	0	51	40	57	72	29
Macello_10	0	49	46	30	42	55	73	44	65	52	0	63	45	72	31
Macello_11	0	33	74	56	49	68	39	41	54	41	64	0	52	55	27
Macello_12	0	52	42	49	32	53	63	77	56	55	42	56	0	30	37
Macello_13	0	45	53	68	42	35	46	39	37	71	74	57	29	0	40
Conceria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CAMION_2	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10	Macello_11	Macello_12	Macello_13	Conceria
Deposito	0	30	40	57	44	43	33	29	34	35	32	43	37	35	0
Macello_1	0	0	47	66	59	51	50	47	42	35	46	33	50	44	35
Macello_2	0	43	0	54	63	58	58	54	56	60	45	53	43	48	51
Macello_3	0	63	51	0	71	64	50	59	41	38	29	53	49	64	44
Macello_4	0	68	70	65	0	69	63	73	62	63	41	47	29	42	50
Macello_5	0	45	61	49	57	0	64	63	75	41	50	66	49	35	31
Macello_6	0	60	70	52	62	78	0	68	61	50	72	38	60	42	52
Macello_7	0	74	64	52	56	47	37	0	27	61	40	41	73	36	62
Macello_8	0	54	51	72	46	66	68	59	0	34	61	53	55	37	76
Macello_9	0	34	63	41	64	41	52	58	37	0	47	38	55	66	57
Macello_10	0	46	43	29	40	53	71	42	63	49	0	60	43	68	57
Macello_11	0	31	72	54	48	66	37	39	51	39	62	0	50	53	45
Macello_12	0	51	40	47	31	51	60	74	54	54	41	54	0	30	41
Macello_13	0	42	51	65	39	33	44	37	36	67	70	55	27	0	34
Conceria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CAMION_3	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10	Macello_11	Macello_12	Macello_13	Conceria
Deposito	0	27	38	53	38	42	32	29	32	34	30	41	36	33	0
Macello_1	0	0	46	63	57	50	50	47	40	35	47	34	50	45	29
Macello_2	0	42	0	52	61	57	59	52	56	61	46	54	44	49	40
Macello_3	0	60	50	0	69	65	50	59	41	39	30	54	49	65	55
Macello_4	0	67	68	60	0	70	64	72	61	64	42	48	30	43	42
Macello_5	0	45	59	50	57	0	62	61	76	42	51	67	50	35	38
Macello_6	0	60	70	50	61	79	0	67	62	51	73	39	61	43	35
Macello_7	0	74	64	50	52	48	38	0	28	62	41	41	74	37	42
Macello_8	0	53	47	71	46	67	69	56	0	35	62	53	55	37	59
Macello_9	0	34	64	41	65	41	53	59	37	0	48	39	56	68	37
Macello_10	0	47	44	29	41	54	72	43	64	50	0	61	44	69	44
Macello_11	0	32	73	55	48	67	38	40	52	40	63	0	51	54	48
Macello_12	0	51	41	48	31	52	61	75	55	54	41	55	0	30	64
Macello_13	0	43	52	66	40	34	45	38	36	68	71	56	28	0	42
Conceria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.2. Tabelle di Elaborazione dati

A.2.2.1. Dati elaborazione Tragitto 1

Nel tragitto è stato aggiunto il Macello_1 tra il Deposito e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Conceria	3450	40	285	105	430	7	66

Inizializzazione tragitto numero 1

Nel tragitto è stato aggiunto il Macello_9 tra il Macello_1 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Conceria	2970	63	389	0	452	9	90

1° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_11 tra il Macello_9 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Conceria	2570	95	529	0	624	12	128

2° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_6 tra il Macello_11 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_6	2230	107	506	0	613	9	140
Conceria	2230	117	670	0	787	14	175

3° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_13 tra il Macello_6 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_6	2230	107	506	0	613	9	140
Macello_13	1960	141	650	0	791	13	185
Conceria	1960	151	789	0	940	18	225

4° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_12 tra il Macello_13 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_6	2230	107	506	0	613	9	140
Macello_13	1960	141	650	0	791	13	185
Macello_12	1410	170	771	0	941	17	214
Conceria	1410	180	878	0	1058	22	251

5° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_5 tra il Macello_12 e la Conceria

Il tempo di fine servizio (1119) presso il Macello_5 è superiore al tempo di chiusura dello stesso (900)

6° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_5 tra il Macello_1 e il Macello_9

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_5	3060	60	247	0	307	5	74
Macello_9	2580	83	420	0	503	7	118
Macello_11	2180	115	546	0	661	10	158
Macello_6	1840	137	693	0	830	12	197
Macello_13	1570	171	837	0	1008	16	242
Macello_12	1020	200	958	0	1158	20	271
Conceria	1020	210	1065	0	1275	25	308

7° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_7 tra il Macello_12 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_5	3060	60	247	0	307	5	74
Macello_9	2580	83	420	0	503	7	118
Macello_11	2180	115	546	0	661	10	158
Macello_6	1840	137	693	0	830	12	197
Macello_13	1570	171	837	0	1008	16	242
Macello_12	1020	200	958	0	1158	20	271
Macello_7	600	219	1130	0	1349	22	348
Conceria	600	229	1271	0	1500	27	385

8° iterazione di assegnazione

A.2.2.2.Dati elaborazione Tragitto 2

Nel tragitto è stato aggiunto il Macello_3 tra il Deposito e la Conceria

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Conceria	1900	77	330	23	430	8	108

Inizializzazione tragitto numero 2

Nel tragitto è stato aggiunto il Macello_2 tra il Macello_3 e la Conceria

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Conceria	1370	107	426	3	536	11	143

1° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_10 tra il Deposito e il Macello_3

Il tempo di fine servizio (925) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

2° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_4 tra il Macello_2 e la Conceria

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Macello_4	1070	117	471	3	591	9	164
Conceria	1070	127	578	3	708	13	206

3° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_10 tra il Deposito e il Macello_3

Il tempo di fine servizio (925) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

4° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_8 tra il Macello_4 e la Conceria

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Macello_4	1070	117	471	3	591	9	164
Macello_8	680	161	586	3	750	13	225
Conceria	680	171	714	3	888	17	284

5° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_10 tra il Macello_8 e la Conceria

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Macello_4	1070	117	471	3	591	9	164
Macello_8	680	161	586	3	750	13	225
Macello_10	60	208	764	3	975	16	287
Conceria	60	218	896	3	1117	20	331

6° iterazione di assegnazione

A.3.TABELLE DEL PROBLEMA CON 20 MACELLI E 5 CAMION

A.3.1.Dati di Input

A.3.1.1.Tempi di viaggio per ogni camion

CAMION_1	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
Deposito	0	125	137	151	134	129	120	90	112	106	149
Macello_1	0	0	145	161	156	122	125	111	137	108	149
Macello_2	0	142	0	150	160	167	156	163	123	175	141
Macello_3	0	158	149	0	168	168	141	128	181	148	124
Macello_4	0	166	167	158	0	171	152	181	114	166	116
Macello_5	0	156	181	145	132	0	120	100	161	173	146
Macello_6	0	112	107	119	137	113	0	115	101	150	161
Macello_7	0	109	111	133	112	100	128	0	98	146	151
Macello_8	0	131	140	129	107	119	170	159	0	121	177
Macello_9	0	103	177	145	170	138	152	149	119	0	133
Macello_10	0	152	138	120	115	143	166	153	174	131	0
Macello_11	0	168	113	157	136	184	147	107	129	130	124
Macello_12	0	134	160	144	115	148	136	172	151	164	170
Macello_13	0	154	137	178	151	118	146	149	103	172	169
Macello_14	0	109	153	121	129	144	109	153	121	129	144
Macello_15	0	111	154	166	142	119	111	154	166	142	119
Macello_16	0	177	144	115	158	154	177	144	115	158	154
Macello_17	0	151	128	159	148	141	151	128	159	148	141
Macello_18	0	170	119	141	114	150	170	119	141	114	150
Macello_19	0	175	148	186	155	119	175	148	186	155	119
Macello_20	0	153	165	148	141	146	153	165	148	141	146
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_1 – prima parte

CAMION_1	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
	Deposito	120	123	141	134	111	154	166	142	119	112
Macello_1	166	139	122	128	177	144	115	158	154	140	160
Macello_2	111	157	152	139	151	128	159	148	141	149	150
Macello_3	154	146	139	153	170	119	141	114	150	117	143
Macello_4	139	112	148	139	175	148	186	155	119	145	130
Macello_5	182	151	116	141	153	165	148	141	146	151	165
Macello_6	143	138	144	128	149	153	109	170	153	102	164
Macello_7	105	167	150	158	92	120	109	153	121	176	141
Macello_8	133	149	101	160	112	139	111	154	166	170	126
Macello_9	126	160	175	136	166	125	177	144	115	161	156
Macello_10	168	123	169	124	131	187	151	128	159	121	149
Macello_11	0	157	160	151	190	116	170	119	141	148	170
Macello_12	154	0	120	144	101	170	175	148	186	165	107
Macello_13	158	121	0	137	116	107	153	165	148	146	139
Macello_14	128	126	143	0	166	142	119	131	125	112	148
Macello_15	131	125	140	138	0	132	162	158	160	151	116
Macello_16	142	139	111	144	128	0	150	140	132	138	144
Macello_17	157	125	177	119	131	120	0	139	142	167	150
Macello_18	146	187	151	154	142	148	150	0	144	149	101
Macello_19	142	116	170	141	157	160	163	168	0	160	175
Macello_20	129	170	175	150	146	162	174	140	144	0	151
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_1 – seconda parte

CAMION_2	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
Deposito	0	130	140	157	144	133	121	92	120	109	153
Macello_1	0	0	147	166	159	123	127	112	139	111	154
Macello_2	0	143	0	154	163	165	156	166	125	177	144
Macello_3	0	163	151	0	171	166	144	131	187	151	128
Macello_4	0	168	170	165	0	170	154	190	116	170	119
Macello_5	0	157	178	147	133	0	120	101	170	175	148
Macello_6	0	115	109	117	135	115	0	116	107	153	165
Macello_7	0	110	113	140	112	102	132	0	99	149	153
Macello_8	0	134	143	132	109	121	170	161	0	123	180
Macello_9	0	105	183	147	175	142	155	153	123	0	135
Macello_10	0	156	141	124	118	147	171	155	177	134	0
Macello_11	0	174	118	161	142	190	151	112	132	134	128
Macello_12	0	137	165	147	118	152	138	177	154	167	175
Macello_13	0	157	140	184	153	119	150	151	104	175	174
Macello_14	0	109	111	133	112	100	121	133	128	159	148
Macello_15	0	131	140	129	107	119	153	144	119	141	114
Macello_16	0	103	177	145	170	138	140	160	148	186	155
Macello_17	0	152	138	120	115	143	149	150	165	148	141
Macello_18	0	168	113	157	136	184	117	143	153	109	170
Macello_19	0	134	160	144	115	148	145	130	180	136	151
Macello_20	0	154	137	178	151	118	151	165	135	131	164
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_2 – prima parte

CAMION_2	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
Deposito	121	129	144	128	126	143	151	140	108	125	0
Macello_1	166	142	119	131	125	126	144	150	120	170	150
Macello_2	115	158	154	142	139	111	154	139	111	154	166
Macello_3	159	148	141	157	125	177	144	125	177	144	115
Macello_4	141	114	150	146	187	151	128	187	151	128	159
Macello_5	186	155	119	142	116	170	119	116	170	119	141
Macello_6	148	141	146	129	170	175	148	170	175	148	186
Macello_7	109	170	153	162	107	153	165	107	153	165	148
Macello_8	136	151	105	162	124	154	146	139	153	122	133
Macello_9	131	164	178	139	116	139	112	148	139	166	140
Macello_10	172	126	171	125	146	182	151	116	141	112	158
Macello_11	0	160	162	154	161	143	138	144	128	156	147
Macello_12	156	0	123	147	151	105	167	150	158	140	113
Macello_13	162	123	0	140	177	133	149	101	160	184	153
Macello_14	141	157	160	0	133	126	160	175	136	145	13
Macello_15	150	146	130	108	0	140	165	137	123	106	168
Macello_16	119	142	146	172	180	0	125	170	169	134	150
Macello_17	146	129	121	116	120	121	0	165	140	128	162
Macello_18	153	162	159	139	160	153	166	0	166	169	124
Macello_19	105	162	145	116	170	155	136	148	0	145	147
Macello_20	178	139	181	152	135	148	177	159	146	0	136
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_2 – seconda parte

CAMION_3	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
	Deposito	0	132	143	160	148	138	121	96	124	115
Macello_1	0	0	146	163	157	122	126	111	138	109	151
Macello_2	0	142	0	152	161	166	157	164	124	176	142
Macello_3	0	160	150	0	169	167	142	129	184	149	125
Macello_4	0	167	168	150	0	171	153	184	115	167	117
Macello_5	0	156	179	146	133	0	121	101	164	174	147
Macello_6	0	113	108	118	136	114	0	115	104	151	163
Macello_7	0	110	112	136	112	102	132	0	99	149	153
Macello_8	0	134	143	132	109	121	170	161	0	123	180
Macello_9	0	105	183	147	175	142	155	153	123	0	135
Macello_10	0	156	141	124	118	147	171	155	177	134	0
Macello_11	0	174	118	161	142	190	151	112	132	134	128
Macello_12	0	137	165	147	118	152	138	177	154	167	175
Macello_13	0	155	138	181	152	119	148	150	103	174	171
Macello_14	0	109	111	133	112	100	121	142	116	170	119
Macello_15	0	131	140	129	107	119	153	129	170	175	148
Macello_16	0	103	177	145	170	138	140	162	107	153	165
Macello_17	0	152	138	120	115	143	149	162	124	154	146
Macello_18	0	168	113	157	136	184	117	139	116	139	112
Macello_19	0	134	160	144	115	148	145	125	146	182	151
Macello_20	0	154	137	178	151	118	151	154	161	143	138
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_3 – prima parte

CAMION_3	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
Deposito	122	133	146	150	144	115	158	154	142	135	0
Macello_1	166	140	121	133	128	159	148	141	157	146	187
Macello_2	112	158	153	144	119	141	114	150	146	142	116
Macello_3	156	147	140	160	148	186	155	119	142	129	170
Macello_4	140	113	149	150	165	148	141	146	129	162	107
Macello_5	184	153	117	143	153	109	170	153	162	162	124
Macello_6	145	139	145	130	180	136	151	105	162	139	116
Macello_7	109	170	153	162	107	153	165	107	153	165	146
Macello_8	136	151	105	162	124	154	146	139	153	122	128
Macello_9	131	164	178	139	116	139	112	148	139	166	177
Macello_10	172	126	171	125	146	182	151	116	141	112	132
Macello_11	0	160	162	154	161	143	138	144	128	156	154
Macello_12	156	0	123	147	151	184	153	119	150	151	104
Macello_13	160	122	0	142	111	133	112	100	121	133	128
Macello_14	138	177	154	0	140	129	107	119	153	144	119
Macello_15	150	151	104	140	0	130	150	165	148	152	130
Macello_16	121	133	128	150	160	0	175	168	133	166	142
Macello_17	153	144	119	122	123	125	0	142	164	170	119
Macello_18	140	160	148	160	121	138	159	0	161	144	127
Macello_19	149	150	165	154	133	162	171	163	0	157	143
Macello_20	117	143	153	147	169	136	140	151	160	0	137
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_3 – seconda parte

CAMION_4	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
	Deposito	0	140	132	120	115	129	107	119	129	144
Macello_1	0	0	152	144	136	145	170	138	142	119	131
Macello_2	0	160	0	139	141	120	115	143	158	154	142
Macello_3	0	128	150	0	160	157	136	184	148	141	157
Macello_4	0	131	140	129	0	144	115	148	114	150	146
Macello_5	0	103	177	145	165	0	170	120	155	119	142
Macello_6	0	152	138	120	129	167	0	146	135	152	144
Macello_7	0	144	115	158	154	142	135	0	146	137	154
Macello_8	0	128	159	148	141	157	146	147	0	155	123
Macello_9	0	119	141	114	150	146	142	151	161	0	170
Macello_10	0	148	186	155	119	142	129	164	153	123	0
Macello_11	0	165	148	141	146	129	162	170	140	130	128
Macello_12	0	153	109	170	153	162	162	174	152	165	172
Macello_13	0	180	136	151	105	162	139	150	103	174	171
Macello_14	0	135	131	164	178	139	125	142	116	170	119
Macello_15	0	183	147	175	142	155	153	129	170	175	148
Macello_16	0	141	124	118	147	171	155	162	107	153	165
Macello_17	0	118	161	142	190	151	112	162	124	154	146
Macello_18	0	166	147	118	152	138	117	144	148	150	151
Macello_19	0	140	184	153	119	150	151	150	165	121	133
Macello_20	0	154	122	169	136	144	157	168	146	153	144
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_4 – prima parte

CAMION_4	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
Deposito	126	143	151	140	108	125	157	178	147	133	0
Macello_1	125	126	144	150	120	170	115	109	117	135	135
Macello_2	139	111	154	139	111	154	110	113	140	112	146
Macello_3	125	177	144	125	177	144	134	143	132	109	142
Macello_4	187	151	128	187	151	128	105	183	147	175	129
Macello_5	116	170	119	116	170	119	156	141	124	118	162
Macello_6	148	150	151	104	140	150	174	118	161	142	162
Macello_7	165	121	133	128	150	165	137	166	147	118	139
Macello_8	146	153	144	119	122	124	157	140	184	153	125
Macello_9	112	140	160	148	160	133	109	111	133	112	153
Macello_10	151	149	150	165	154	168	131	140	129	107	155
Macello_11	0	165	170	126	175	148	103	177	145	170	112
Macello_12	155	0	146	172	153	165	152	138	120	115	117
Macello_13	160	162	0	133	154	146	168	113	157	136	151
Macello_14	138	114	150	0	150	151	134	160	144	115	133
Macello_15	150	155	119	140	0	133	154	137	178	151	144
Macello_16	121	141	146	165	148	0	133	156	119	138	114
Macello_17	153	170	153	153	109	171	0	128	148	150	155
Macello_18	104	151	105	180	136	163	162	0	165	121	141
Macello_19	128	164	178	135	131	121	160	159	0	145	144
Macello_20	119	175	142	183	147	158	145	131	129	0	162
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_4 – seconda parte

CAMION_5	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
	Deposito	0	140	132	120	115	129	107	119	129	144
Macello_1	0	0	152	144	136	145	170	138	142	119	131
Macello_2	0	160	0	139	141	120	115	143	158	154	142
Macello_3	0	128	150	0	160	157	136	184	177	141	157
Macello_4	0	131	140	129	0	144	115	148	114	150	146
Macello_5	0	103	177	145	165	0	170	122	155	119	142
Macello_6	0	152	138	120	129	167	0	146	135	152	144
Macello_7	0	144	115	158	154	154	135	0	146	137	154
Macello_8	0	165	159	148	141	157	146	147	0	155	123
Macello_9	0	119	141	114	150	146	144	151	161	0	170
Macello_10	0	148	186	155	119	142	129	164	153	123	0
Macello_11	0	121	148	141	146	129	160	170	140	130	128
Macello_12	0	155	120	142	153	162	162	174	152	165	172
Macello_13	0	180	136	150	105	162	139	150	112	186	171
Macello_14	0	162	131	164	178	139	125	142	116	170	119
Macello_15	0	178	147	175	142	155	153	129	123	175	148
Macello_16	0	141	124	123	147	171	155	162	107	153	165
Macello_17	0	118	161	142	181	151	112	162	124	154	146
Macello_18	0	166	144	129	152	138	117	144	148	150	151
Macello_19	0	145	184	153	119	155	151	123	165	121	133
Macello_20	0	154	122	169	136	144	157	167	146	153	144
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_5 – prima parte

CAMION_5	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
Deposito	126	143	119	141	114	113	108	118	136	152	0
Macello_1	125	126	148	186	155	110	112	136	113	167	168
Macello_2	139	111	165	148	141	132	141	130	108	156	179
Macello_3	125	177	153	109	170	104	179	146	172	113	108
Macello_4	187	151	180	136	151	154	139	121	116	110	112
Macello_5	116	170	135	131	164	170	115	159	139	132	141
Macello_6	148	150	183	147	175	135	162	145	116	104	179
Macello_7	165	121	141	124	118	165	138	181	152	154	139
Macello_8	146	153	118	161	142	109	111	133	112	170	115
Macello_9	112	140	166	147	118	131	140	129	107	135	162
Macello_10	151	149	140	184	153	103	177	145	170	155	138
Macello_11	0	165	154	122	169	152	138	120	115	109	111
Macello_12	155	0	110	120	130	168	113	157	136	131	140
Macello_13	160	162	0	122	160	134	160	144	115	103	177
Macello_14	138	114	162	0	140	154	137	178	151	152	138
Macello_15	150	156	135	109	0	152	160	170	140	150	120
Macello_16	121	141	114	180	150	0	156	166	176	136	136
Macello_17	153	170	140	157	146	160	0	175	134	165	146
Macello_18	104	151	144	110	123	168	168	0	147	159	141
Macello_19	128	164	150	120	130	131	147	156	0	165	135
Macello_20	119	175	162	144	120	133	144	145	126	0	126
Conceria	0	0	0	0	0	0	0	0	0	0	0

Tempi di viaggio Camion_5 – seconda parte

A.3.1.2.Costi di viaggio per ogni camion

CAMION_1	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
Deposito	0	25	37	51	34	41	33	28	31	33	29
Macello_1	0	0	45	61	56	49	51	46	39	36	48
Macello_2	0	42	0	50	60	56	60	51	57	63	47
Macello_3	0	58	49	0	68	64	51	60	42	40	31
Macello_4	0	66	67	58	0	71	66	72	61	66	43
Macello_5	0	45	58	49	58	0	61	59	77	44	53
Macello_6	0	59	71	49	61	80	0	67	63	52	75
Macello_7	0	75	65	49	51	49	39	0	29	63	42
Macello_8	0	53	45	71	47	69	70	54	0	36	64
Macello_9	0	35	65	42	67	43	54	61	38	0	51
Macello_10	0	49	46	30	42	55	73	44	65	52	0
Macello_11	0	33	74	56	49	68	39	41	54	41	64
Macello_12	0	52	42	49	32	53	63	77	56	55	42
Macello_13	0	45	53	68	42	35	46	39	37	71	74
Macello_14	0	44	52	49	65	60	44	23	35	40	65
Macello_15	0	66	58	71	45	38	71	38	66	44	43
Macello_16	0	41	54	64	41	43	39	34	32	63	45
Macello_17	0	24	52	55	35	41	62	30	40	50	71
Macello_18	0	32	73	28	50	60	71	42	55	30	29
Macello_19	0	41	39	72	60	51	42	36	61	69	54
Macello_20	0	39	54	44	41	25	38	67	45	37	33
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_1 – prima parte

CAMION_1	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
Deposito	40	35	32	36	42	33	34	29	17	30	0
Macello_1	35	51	46	28	50	48	29	35	32	37	41
Macello_2	55	44	51	39	29	25	34	38	53	26	39
Macello_3	55	50	67	53	40	31	53	50	69	53	50
Macello_4	50	31	44	39	66	44	50	31	44	39	45
Macello_5	69	52	36	37	44	53	69	52	36	37	41
Macello_6	40	62	45	36	52	70	41	62	45	36	35
Macello_7	42	76	38	41	63	42	42	77	38	41	37
Macello_8	54	57	38	58	36	64	54	58	38	58	26
Macello_9	40	57	72	36	42	51	40	57	73	39	29
Macello_10	63	45	72	26	52	43	63	44	72	27	31
Macello_11	0	52	55	49	41	64	37	52	61	48	27
Macello_12	56	0	30	40	55	42	56	36	30	40	37
Macello_13	57	29	0	42	53	62	64	70	34	38	40
Macello_14	72	32	63	0	60	72	55	50	29	48	35
Macello_15	66	38	36	71	0	72	47	27	35	54	45
Macello_16	49	35	44	50	60	0	61	46	34	53	41
Macello_17	48	28	30	72	67	40	0	33	69	41	29
Macello_18	33	35	62	61	66	51	54	0	35	40	51
Macello_19	45	31	44	47	72	38	45	39	0	39	27
Macello_20	71	69	71	43	47	33	31	29	65	0	37
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_1 – seconda parte

CAMION_2	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
Deposito	0	30	40	57	44	43	33	29	34	35	32
Macello_1	0	0	47	66	59	51	50	47	42	35	46
Macello_2	0	43	0	54	63	58	58	54	56	60	45
Macello_3	0	63	51	0	71	64	50	59	41	38	29
Macello_4	0	68	70	65	0	69	63	73	62	63	41
Macello_5	0	45	61	49	57	0	64	63	75	41	50
Macello_6	0	60	70	52	62	78	0	68	61	50	72
Macello_7	0	74	64	52	56	47	37	0	27	61	40
Macello_8	0	54	51	72	46	66	68	59	0	34	61
Macello_9	0	34	63	41	64	41	52	58	37	0	47
Macello_10	0	46	43	29	40	53	71	42	63	49	0
Macello_11	0	31	72	54	48	66	37	39	51	39	62
Macello_12	0	51	40	47	31	51	60	74	54	54	41
Macello_13	0	42	51	65	39	33	44	37	36	67	70
Macello_14	0	31	33	29	40	35	32	36	42	33	34
Macello_15	0	39	36	48	35	51	46	28	50	48	29
Macello_16	0	57	63	47	55	44	51	39	29	25	34
Macello_17	0	42	40	31	55	50	67	53	40	31	53
Macello_18	0	61	66	43	50	31	44	39	66	44	50
Macello_19	0	77	44	53	69	52	36	37	44	53	69
Macello_20	0	63	52	75	40	62	45	36	52	70	41
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_2 – prima parte

CAMION_2	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
Deposito	43	37	35	40	41	40	39	35	29	31	0
Macello_1	33	50	44	51	47	54	44	49	40	40	35
Macello_2	53	43	48	42	30	54	49	65	55	35	51
Macello_3	53	49	64	57	42	48	30	43	42	55	44
Macello_4	47	29	42	46	51	67	50	35	38	55	50
Macello_5	66	49	35	39	73	39	61	43	35	50	31
Macello_6	38	60	42	35	41	41	74	37	42	69	52
Macello_7	41	73	36	43	62	53	55	37	59	40	62
Macello_8	53	55	37	59	48	39	56	68	37	42	76
Macello_9	38	55	66	37	35	46	33	50	44	54	57
Macello_10	60	43	68	28	60	45	53	43	48	40	57
Macello_11	0	50	53	52	38	29	53	49	64	63	45
Macello_12	54	0	30	41	63	41	47	29	42	30	41
Macello_13	55	27	0	44	41	50	66	45	35	47	34
Macello_14	45	60	55	0	50	72	38	60	42	47	54
Macello_15	32	30	40	51	0	50	40	44	55	30	54
Macello_16	33	15	22	39	33	0	51	30	60	42	48
Macello_17	29	32	34	30	41	45	0	41	61	51	67
Macello_18	47	40	35	47	34	60	50	0	40	73	39
Macello_19	52	56	61	47	54	70	60	41	0	41	41
Macello_20	59	41	39	30	54	50	50	46	32	0	40
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_2 – seconda parte

CAMION_3	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
	Deposito	0	27	38	53	38	42	32	29	32	34
Macello_1	0	0	46	63	57	50	50	47	40	35	47
Macello_2	0	42	0	52	61	57	59	52	56	61	46
Macello_3	0	60	50	0	69	65	50	59	41	39	30
Macello_4	0	67	68	60	0	70	64	72	61	64	42
Macello_5	0	45	59	50	57	0	62	61	76	42	51
Macello_6	0	60	70	50	61	79	0	67	62	51	73
Macello_7	0	74	64	50	52	48	38	0	28	62	41
Macello_8	0	53	47	71	46	67	69	56	0	35	62
Macello_9	0	34	64	41	65	41	53	59	37	0	48
Macello_10	0	47	44	29	41	54	72	43	64	50	0
Macello_11	0	32	73	55	48	67	38	40	52	40	63
Macello_12	0	51	41	48	31	52	61	75	55	54	41
Macello_13	0	43	52	66	40	34	45	38	36	68	71
Macello_14	0	31	33	29	40	35	32	36	40	34	30
Macello_15	0	39	36	48	35	51	46	28	45	35	47
Macello_16	0	35	61	35	42	54	50	60	44	61	47
Macello_17	0	29	34	35	32	43	37	35	40	39	30
Macello_18	0	47	42	35	46	33	50	44	51	64	42
Macello_19	0	45	56	60	45	53	43	48	42	42	51
Macello_20	0	59	41	38	29	53	49	64	57	51	73
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_3 – prima parte

CAMION_3	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
	Deposito	41	36	33	40	74	64	52	56	47	37
Macello_1	34	50	45	29	54	51	72	46	66	68	29
Macello_2	54	44	49	40	34	63	41	64	41	52	40
Macello_3	54	49	65	55	46	43	29	40	53	71	55
Macello_4	48	30	43	42	31	72	54	48	76	37	42
Macello_5	67	50	35	38	51	40	47	31	51	60	38
Macello_6	39	61	43	35	42	51	65	39	33	44	35
Macello_7	41	74	37	42	31	33	29	40	35	32	42
Macello_8	53	55	37	59	39	36	48	35	51	46	59
Macello_9	39	56	68	37	57	63	47	55	44	51	37
Macello_10	61	44	69	27	42	40	31	55	50	67	44
Macello_11	0	51	54	50	61	66	43	50	31	44	48
Macello_12	55	0	30	41	77	44	53	69	52	36	64
Macello_13	56	28	0	43	63	52	75	40	62	45	42
Macello_14	41	36	42	0	20	53	30	45	44	41	35
Macello_15	34	50	40	44	0	61	43	53	54	51	42
Macello_16	54	44	50	41	38	0	44	52	45	33	55
Macello_17	54	49	60	44	69	30	0	44	33	24	60
Macello_18	48	30	45	60	36	44	65	0	68	44	61
Macello_19	67	50	31	30	58	53	47	31	0	30	40
Macello_20	39	61	59	47	30	65	71	34	22	0	39
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_3 – seconda parte

CAMION_4	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
	Deposito	0	27	38	53	38	42	32	29	32	34
Macello_1	0	0	46	63	57	55	50	47	40	55	47
Macello_2	0	42	0	52	61	57	59	52	56	61	47
Macello_3	0	60	50	0	69	65	55	59	55	39	30
Macello_4	0	67	68	60	0	70	64	72	61	64	42
Macello_5	0	45	59	50	57	0	62	61	76	42	51
Macello_6	0	60	70	50	61	79	0	67	62	51	73
Macello_7	0	74	64	50	52	48	38	0	28	55	41
Macello_8	0	53	47	71	46	55	69	56	0	35	62
Macello_9	0	34	64	41	65	41	53	59	37	0	48
Macello_10	0	47	44	29	41	54	72	43	64	50	0
Macello_11	0	32	73	55	48	55	38	55	52	40	63
Macello_12	0	51	41	48	31	52	61	75	55	54	41
Macello_13	0	43	52	66	40	34	45	38	36	68	71
Macello_14	0	31	33	29	40	35	32	55	40	34	30
Macello_15	0	39	36	48	35	51	46	28	45	37	47
Macello_16	0	35	61	35	42	54	50	60	44	61	47
Macello_17	0	29	34	35	32	43	37	35	40	39	30
Macello_18	0	47	42	35	46	33	50	44	51	64	42
Macello_19	0	45	56	60	45	53	43	48	42	42	51
Macello_20	0	59	41	38	29	53	49	64	38	51	73
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_4 – prima parte

CAMION_4	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
	Deposito	41	36	33	40	74	64	52	56	47	37
Macello_1	34	50	44	29	54	51	33	46	66	68	29
Macello_2	54	44	49	40	34	63	41	64	41	52	40
Macello_3	55	49	65	55	46	43	29	40	53	71	55
Macello_4	48	30	43	55	31	72	54	48	76	37	42
Macello_5	67	50	35	38	51	40	47	33	33	60	38
Macello_6	39	61	43	35	42	51	65	33	33	44	35
Macello_7	55	74	37	55	31	33	29	40	35	32	42
Macello_8	53	55	37	59	39	36	48	35	51	46	59
Macello_9	39	56	68	37	57	63	47	55	44	51	37
Macello_10	61	44	69	27	42	40	31	55	50	32	44
Macello_11	0	51	54	50	61	66	43	50	31	44	48
Macello_12	55	0	30	41	77	44	53	31	52	36	64
Macello_13	56	28	0	43	63	52	75	40	62	45	42
Macello_14	41	36	42	0	20	53	30	36	44	41	35
Macello_15	34	55	40	44	0	61	43	53	54	51	42
Macello_16	54	44	50	41	38	0	44	52	45	33	55
Macello_17	34	33	60	44	69	30	0	37	33	24	60
Macello_18	48	55	45	22	36	44	65	0	68	44	61
Macello_19	67	50	31	30	58	53	47	31	0	30	40
Macello_20	39	61	39	47	31	65	71	34	22	0	39
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_4 – seconda parte

CAMION_5	Deposito	Macello_1	Macello_2	Macello_3	Macello_4	Macello_5	Macello_6	Macello_7	Macello_8	Macello_9	Macello_10
	Deposito	0	30	40	57	44	43	33	29	34	35
Macello_1	0	0	47	66	59	51	50	47	42	35	46
Macello_2	0	43	0	54	63	65	58	54	56	60	45
Macello_3	0	63	51	0	71	64	50	59	41	38	29
Macello_4	0	68	70	65	0	69	63	73	56	63	41
Macello_5	0	45	65	49	57	0	64	63	75	41	50
Macello_6	0	60	70	52	62	78	0	68	61	50	72
Macello_7	0	74	64	52	56	47	37	0	27	61	40
Macello_8	0	54	51	72	46	66	68	59	0	34	61
Macello_9	0	34	65	41	64	41	52	58	37	0	47
Macello_10	0	46	43	29	40	65	71	42	63	49	0
Macello_11	0	31	72	66	48	76	37	37	51	39	62
Macello_12	0	51	40	47	31	51	60	74	54	54	41
Macello_13	0	42	44	65	39	33	44	37	65	67	70
Macello_14	0	65	33	29	40	35	66	36	42	33	43
Macello_15	0	39	36	48	35	51	46	28	50	48	29
Macello_16	0	57	63	47	55	44	51	39	39	25	34
Macello_17	0	42	40	31	55	50	67	53	40	31	53
Macello_18	0	61	65	43	50	31	65	39	66	44	50
Macello_19	0	77	55	53	69	52	36	37	44	53	69
Macello_20	0	63	52	75	40	62	45	36	52	70	41
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_5 – prima parte

CAMION_5	Macello_11	Macello_12	Macello_13	Macello_14	Macello_15	Macello_16	Macello_17	Macello_18	Macello_19	Macello_20	Conceria
	Deposito	43	37	35	40	41	40	39	35	29	31
Macello_1	33	50	44	51	47	54	44	49	40	40	35
Macello_2	53	43	48	42	30	54	49	65	55	35	51
Macello_3	53	49	64	57	42	48	30	43	42	55	44
Macello_4	47	54	58	46	51	67	50	35	38	55	50
Macello_5	66	45	35	54	73	39	61	43	35	54	31
Macello_6	38	60	42	35	54	41	74	37	42	69	52
Macello_7	41	73	36	55	62	57	55	37	59	40	62
Macello_8	53	55	37	59	48	39	56	68	37	42	76
Macello_9	38	55	66	37	35	46	33	50	54	54	57
Macello_10	60	43	68	28	60	45	53	43	48	40	57
Macello_11	0	50	53	52	38	56	53	49	64	63	45
Macello_12	54	0	30	41	63	41	47	29	42	30	54
Macello_13	55	27	0	44	41	50	66	45	35	54	34
Macello_14	45	60	55		50	72	38	48	42	47	54
Macello_15	32	30	40	51	0	50	40	44	55	30	54
Macello_16	33	15	22	39	33	0	51	30	60	42	48
Macello_17	29	44	34	48	41	45	0	41	61	54	67
Macello_18	47	54	35	47	34	60	50	0	40	73	39
Macello_19	52	56	61	47	54	70	60	41	0	41	41
Macello_20	59	61	39	30	54	50	50	46	32	0	40
Conceria	0	0	0	0	0	0	0	0	0	0	0

Costi di viaggio Camion_5 – seconda parte

A.3.2. Tabelle di Elaborazione dati

A.3.2.1. Dati elaborazione Tragitto 1

Nel tragitto è stato aggiunto il Macello_1 tra il Deposito e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Conceria	3450	40	285	105	430	7	66

Inizializzazione tragitto numero 1

Nel tragitto è stato aggiunto il Macello_9 tra il Macello_1 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Conceria	2970	63	389	0	452	9	90

1° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_11 tra il Macello_9 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Conceria	2570	95	529	0	624	12	128

2° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_17 tra il Macello_11 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_17	2150	126	529	0	655	10	138
Conceria	2150	136	679	0	815	15	167

3° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_12 tra il Macello_17 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_17	2150	126	529	0	655	10	138
Macello_12	1600	155	654	0	809	14	166
Conceria	1600	165	761	0	926	19	203

4° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_13 tra il Macello_12 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_17	2150	126	529	0	655	10	138
Macello_12	1600	155	654	0	809	14	166
Macello_13	1330	189	774	0	963	18	196
Conceria	1330	199	913	0	1112	23	236

5° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_6 tra il Macello_13 e la Conceria

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_17	2150	126	529	0	655	10	138
Macello_12	1600	155	654	0	809	14	166
Macello_13	1330	189	774	0	963	18	196
Macello_6	990	211	920	0	1131	20	242
Conceria	990	221	1084	0	1305	25	277

6° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_15 tra il Macello_12 e il Macello_13

Il tempo di fine servizio (945) presso il Macello_15 è superiore al tempo di chiusura dello stesso (820)

7° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_15 tra il Macello_6 e la Conceria

Il tempo di fine servizio (1315) presso il Macello_15 è superiore al tempo di chiusura dello stesso (820)

8° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_15 tra il Macello_11 e il Macello_17

Camion_1	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	10	0	0	0	0	0
Macello_1	3450	30	125	0	155	2	25
Macello_9	2970	53	233	0	286	4	61
Macello_11	2570	85	359	0	444	7	101
Macello_15	2390	120	549	0	669	11	142
Macello_17	1970	161	711	0	872	14	189
Macello_12	1420	190	836	0	1026	18	217
Macello_13	1150	224	956	0	1180	22	247
Macello_6	810	246	1102	0	1348	24	293
Conceria	810	256	1266	0	1522	29	328

9° iterazione di assegnazione

A.3.2.2.Dati elaborazione Tragitto 2

Nel tragitto è stato aggiunto il Macello_5 tra il Deposito e la Conceria

Camion_4	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	20	0	0	0	0	0
Macello_5	3310	51	129	61	241	4	42
Conceria	3310	61	291	139	491	9	80

Inizializzazione tragitto numero 2

Nel tragitto è stato aggiunto il Macello_19 tra il Macello_5 e la Conceria

Camion_4	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	20	0	0	0	0	0
Macello_5	3310	51	129	61	241	4	42
Macello_19	2800	82	253	157	492	7	75
Conceria	2800	92	397	157	646	12	115

1° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_20 tra il Macello_19 e la Conceria

Camion_4	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	20	0	0	0	0	0
Macello_5	3310	51	129	61	241	4	42
Macello_19	2800	82	253	157	492	7	75
Macello_20	2520	103	398	157	658	9	105
Conceria	2520	113	560	157	830	14	144

2° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_16 tra il Macello_5 e il Macello_19

Camion_4	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	20	0	0	0	0	0
Macello_5	3310	51	129	61	241	4	42
Macello_16	2960	74	248	61	383	6	82
Macello_19	2450	105	367	61	533	9	127
Macello_20	2170	126	512	61	699	11	157
Conceria	2170	136	674	61	871	16	196

3° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_16 e il Macello_19

Il tempo di fine servizio (535) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

4° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_18 tra il Macello_20 e la Conceria

Camion_4	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	20	0	0	0	0	0
Macello_5	3310	51	129	61	241	4	42
Macello_16	2960	74	248	61	383	6	82
Macello_19	2450	105	367	61	533	9	127
Macello_20	2170	126	512	61	699	11	157
Macello_18	1790	156	643	61	860	13	191
Conceria	1790	166	784	61	1011	18	252

5° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_18 e la Conceria

Il tempo di fine servizio (1012) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

6° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_16 e il Macello_19

Il tempo di fine servizio (535) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

7° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_8 tra il Macello_18 e la Conceria

Camion_4	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	20	0	0	0	0	0
Macello_5	3310	51	129	61	241	4	42
Macello_16	2960	74	248	61	383	6	82
Macello_19	2450	105	367	61	533	9	127
Macello_20	2170	126	512	61	699	11	157
Macello_18	1790	156	643	61	860	13	191
Macello_8	1400	199	791	61	1051	16	242
Conceria	1400	209	916	61	1186	21	301

8° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_7 tra il Macello_8 e la Conceria

Camion_4	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	3700	20	0	0	0	0	0
Macello_5	3310	51	129	61	241	4	42
Macello_16	2960	74	248	61	383	6	82
Macello_19	2450	105	367	61	533	9	127
Macello_20	2170	126	512	61	699	11	157
Macello_18	1790	156	643	61	860	13	191
Macello_8	1400	199	791	61	1051	16	242
Macello_7	980	222	938	61	1221	19	298
Conceria	980	232	1077	61	1370	24	340

9° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_16 e il Macello_19

Il tempo di fine servizio (535) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

10° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_20 e il Macello_18

Il tempo di fine servizio (902) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

11° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_7 e la Conceria

Il tempo di fine servizio (1413) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

12° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_5 e il Macello_16

Il tempo di fine servizio (1255) presso il Macello_8 è superiore al tempo di chiusura dello stesso (1200)

13° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Deposito e il Macello_5

Il tempo di fine servizio (1212) presso il Macello_8 è superiore al tempo di chiusura dello stesso (1200)

14° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_18 e il Macello_8

Il tempo di fine servizio (1012) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

15° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_19 e il Macello_20

Il tempo di fine servizio (720) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

16° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_8 e il Macello_7

Il tempo di fine servizio (1233) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

17° iterazione di assegnazione

A.3.2.3.Dati elaborazione Tragitto 3

Nel tragitto è stato aggiunto il Macello_14 tra il Deposito e la Conceria

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_14	2500	58	150	130	338	4	40
Conceria	2500	68	269	223	560	8	75

Inizializzazione tragitto numero 3

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_14 e la Conceria

Il tempo di fine servizio (508) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

1° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_2 tra il Deposito e il Macello_14

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_2	2170	60	143	207	410	3	38
Macello_14	1970	88	287	207	582	7	78
Conceria	1970	98	406	207	711	11	113

2° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_3 tra il Macello_14 e la Conceria

Il tempo di fine servizio (752) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

3° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_3 tra il Deposito e il Macello_2

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Macello_14	1170	125	454	3	582	11	143
Conceria	1170	135	573	3	711	15	178

4° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_4 tra il Macello_14 e la Conceria

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Macello_14	1170	125	454	3	582	11	143
Macello_4	870	145	566	3	714	13	183
Conceria	870	155	673	3	831	17	225

5° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_10 tra il Deposito e il Macello_3

Il tempo di fine servizio (928) presso il Macello_3 è superiore al tempo di chiusura dello stesso (490)

6° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_10 tra il Macello_3 e il Macello_2

Il tempo di fine servizio (1110) presso il Macello_14 è superiore al tempo di chiusura dello stesso (650)

7° iterazione di assegnazione

Nel tragitto NON è stato possibile aggiungere il Macello_10 tra il Macello_2 e il Macello_14

Il tempo di fine servizio (923) presso il Macello_14 è superiore al tempo di chiusura dello stesso (650)

8° iterazione di assegnazione

Nel tragitto è stato aggiunto il Macello_10 tra il Macello_4 e la Conceria

Camion_3	CAPACITA'	TEMPO CARICO	TEMPO VIAGGIO	TEMPO ATTESA	TEMPO FINE SERVIZIO	COSTO CARICO	COSTO VIAGGIO
Deposito	2700	30	0	0	0	0	0
Macello_3	1900	67	160	0	227	4	53
Macello_2	1370	97	310	3	410	7	103
Macello_14	1170	125	454	3	582	11	143
Macello_4	870	145	566	3	714	13	183
Macello_10	250	192	683	3	878	16	225
Conceria	250	202	815	3	1020	20	269

9° iterazione di assegnazione

RINGRAZIAMENTI

Desidero ricordare tutti coloro che mi hanno aiutato nella stesura della tesi con suggerimenti, critiche ed osservazioni: a loro va la mia gratitudine, anche se a me spetta la responsabilità per ogni errore contenuto.

Ringrazio anzitutto il mio relatore, professor Giorgio Romanin Jacur, per avermi dato questa opportunità e proposto un tema così interessante, per i preziosi insegnamenti e per la disponibilità infinita.

Rivolgo estrema gratitudine alla mia famiglia. In particolare sono profondamente debitore ai miei genitori, Giuseppina e Luigi, che hanno sempre assecondato le mie scelte, esaudito i miei desideri, permesso di realizzare i miei sogni e, soprattutto, contribuito a formare la persona che sono.

Ritengo doveroso ringraziare i miei fratelli, Massimo, Francesco, Alberto, e le loro fidanzate, perché mi sono stati accanto in questo lungo percorso, per il supporto e l'allegria che spesso ha distolto la mia attenzione dai problemi, scacciando lo scoraggiamento e la paura.

Ringrazio Chiara, la mia fidanzata, senza la quale niente sarebbe lo stesso. La determinazione, la complicità e l'impegno messi in campo in questo periodo, le fanno onore, assieme ad una dose illimitata di pazienza e comprensione.

Infine, ringrazio tutte le persone che non ho nominato esplicitamente ma che hanno avuto un ruolo importante nella mia vita: i familiari di Chiara, i parenti più stretti, gli amici, i compagni di università e gli ex-colleghi di lavoro. Se sono riuscito a raggiungere questo traguardo è anche grazie a loro, per le esperienze vissute, l'incoraggiamento ed i consigli ricevuti. I ricordi di tutti loro rimarranno impressi in maniera indelebile nel mio cuore.

