

Capitolo 4

La classe LetturaDati

Oggetto di questo capitolo saranno i dati in input nel modello matematico. Le informazioni sui dati concreti del problema, quali domande ed offerte di prodotti, stock iniziale all'inizio della giornata produttiva, orari di inizio e di fine della giornata lavorativa, e anche durata dei viaggi tra origini e destinazioni, con i relativi costi di viaggio carico e scarico, ci sono state inviate dal Dipartimento di Ingegneria Industriale dell'Università del Cile.

L'elaborazione che è stata necessaria per portare questi dati dalla forma grezza, caratteristica di informazioni raccolte sul "campo", ad una forma raffinata indispensabile per il modello matematico ha costituito una fase cruciale del nostro lavoro; in particolare è stata implementata la classe LetturaDati che racchiude al suo interno tali elaborazioni. Per consentire una migliore comprensione delle funzionalità di questa classe verranno illustrati, in maniera sufficientemente dettagliata, i dati nella forma in cui ci sono pervenuti; seguirà la descrizione dei vari procedimenti quali, ad esempio, il calcolo del tempo medi di caricamento e scaricamento, il calcolo degli intervalli di offerta alle origini, ed il calcolo degli intervalli in cui c'è disponibilità di attrezzature per lo scaricamento del legname a destinazione.

4.1 Descrizione dettagliata delle istanze

Le istanze del problema ci sono state inviate dal Dipartimento di Ingegneria Industriale dell'Università del Cile.

Ogni istanza è costituita da una directory contenente i seguenti file :

- pro.exp : Informazioni sui Prodotti
- des.exp : Informazioni sulle Destinazioni
- equ.exp : Informazioni sulle Equipe di scarico a destinazione
- ori.exp : Informazioni sulle Origini

tip.exp	: Informazioni sui tipi di Camion
orides.exp	: Tempi di viaggio carico e tempi di viaggio scarico tra Origini e Destinazioni
oripro.exp	: Offerta di prodotto e tempo di carico per ogni Origine
despro.exp	: Domanda di prodotto per ogni Destinazione
equipro.exp	: Tempo di scarico per ogni equipe di scarico a Destinazione
tipro.exp	: Capacità di carico per ogni tipo di camion

La notazione utilizzata nei file è la seguente:

C X 0 = stringa di X caratteri (es. C 8 0 => ABCDEFGH)

N X Y = numero di X cifre con Y cifre decimali (es. N 8 3 => 1234.567)

- File **pro.exp** (**Prodotti**)

nome	tipo	descrizione
PROCOD	C 8 0	Codice del prodotto

- File **des.exp** (**Destinazioni**)

Nome	Tipo	Descrizione
DESCOD	C 8 0	Codice della destinazione
DESDES	C 25 0	Descrizione della destinazione

- File **equ.exp** (**Equipe**)

Nome	Tipo	Descrizione
EQUCOD	C 8 0	Codice dell'equipe
EQUDES	C 8 0	Codice della destinazione relativa
EQUINI	N 5 2	Ora inizio giornata lavorativa
EQUDUR	N 5 2	Durata giornata lavorativa
EQUSTR	N 5 2	Durata dello straordinario
EQUTU1	N 5 2	Ora inizio primo turno
EQUDU1	N 5 2	Durata primo turno
EQUTU2	N 5 2	Ora inizio secondo turno
EQUDU2	N 5 2	Durata secondo turno

- File **ori.exp** (**Origini**)

Nome	Tipo	Descrizione
ORICOD	C 8 0	Codice dell'origine
ORIDES	C 25 0	Descrizione dell'origine
ORIRES	C 8 0	Codice del responsabile dell'origine
ORIREG	N 5 2	Codice del registro interno
ORIMIN	N 5 2	Ora inizio 1 della giornata lavorativa
ORIMAX	N 5 2	Ora inizio 2 della giornata lavorativa
ORINOR	N 5 2	Durata della giornata di lavoro
ORISTR	N 5 2	Durata dello straordinario
ORIINI	N 5 2	Ora di inizio della produzione
ORIDUR	N 5 2	Durata della produzione
ORIIP1	N 5 2	Ora di inizio pausa 1
ORIDP1	N 5 2	Durata pausa 1
ORIIP2	N 5 2	Ora di inizio pausa 2
ORIDP2	N 5 2	Durata pausa 2
ORIURG	N 1 0	Urgenza di carico

- File **tip.exp** (Tipi di camion)

Nome	Tipo	Descrizione
TIPCOD	C 8 0	Codice del tipo di camion
TIPCVS	N 5 0	Costo orario viaggio scarico
TIPCVC	N 5 0	Costo orario viaggio carico
TIPCFI	N 5 0	Costo fisso orario

- File **orides.exp** (Tempi di viaggio)

Nome	Tipo	Descrizione
ORIDESORI	C 8 0	Codice dell'origine
ORIDESDE	C 8 0	Costo della destinazione
ORIDESCA	N 5 2	Durata viaggio scarico
ORIDESSC	N 5 2	Durata viaggio carico

- File **oripro.exp** (offerta di prodotto e tempo di carico)

Nome	Tipo	Descrizione
ORIPROOR	C 8 0	Codice dell'origine
ORIPROPR	C 8 0	Codice del prodotto
ORIPROST	N 5 0	Stock iniziale (m^3)
ORIPROPG	N 5 0	Produzione giornaliera
ORIPRODU	N 5 2	Durata di carico
ORIPROUR	N 5 2	Urgenza di carico

- File **despro.exp** (domanda di prodotto)

Nome	Tipo	Descrizione
DESPRODE	C 8 0	Codice della destinazione
DESPROPR	C 8 0	Codice del prodotto
DESPRODO	N 5 2	Domanda di prodotto
DESPROUR	N 5 2	Urgenza di scarico

- File **equipro.exp** (tempi di scaricamento)

Nome	Tipo	Descrizione
EQUPROEQ	C 8 0	Codice dell'equipe
EQUPROPR	C 8 0	Codice del prodotto
EQUPROSC	N 5 2	Durata di scarico

- File **tipro.exp** (capacità di carico per tipo di camion)

Nome	Tipo	Descrizione
TIPPROTIP	C 8 0	Codice del tipo di camion
TIPPROPR	C 8 0	Codice del prodotto
TIPPROVO	N 5 2	Capacità di carico

Per maggiore chiarezza nell'appendice A sono riportati alcuni frammenti dei file che sono stati descritti sinora.

4.2 Le strutture della classe LetturaDati

In base a quanto visto nel precedente paragrafo risulta chiaro che sono necessarie diverse elaborazioni per costruire, a partire dai dati “grezzi”, i dati sui quali costruire il modello presentato nel Cap.2. Ad esempio è necessaria l’indicizzazione dei prodotti, dei tipi di camion, delle varie origini etc.., in modo tale da avere accesso diretto a questi dati.

La realizzazione delle diverse strutture dati di questa classe ha avuto lo scopo, oltre che di consentire una maggiore funzionalità nell’accedere direttamente ai dati, anche e soprattutto di semplificare il calcolo di quei dati che identificano il grafo associato al problema. Ad esempio, la determinazione dei deficit dei nodi origine è legata al modo in cui si è scelto di suddividere la produzione giornaliera nei vari intervalli temporali, mentre i deficit ai nodi destinazione sono legati alla ripartizione della domanda secondo criteri che descriveremo in seguito. Tali criteri sono implementati nella classe LetturaDati e trasparenti invece alle altre classi che compongono il codice complessivo. In questo modo, cambiamenti nei criteri comportano solo modifiche a LetturaDati, e non alle altre classi. Ciò è stato importante nella fase di sperimentazione computazionale, come vedremo nel Cap.7.

Al costruttore della classe LetturaDati viene passato come parametro il nome della directory contenente l’istanza. Il costruttore è la parte più importante di questa classe: è qui, infatti, che vengono create le strutture dati, ed eseguiti tutti i calcoli di cui si è accennato. Andiamo ora ad analizzare singolarmente le strutture dati implementate nella classe.

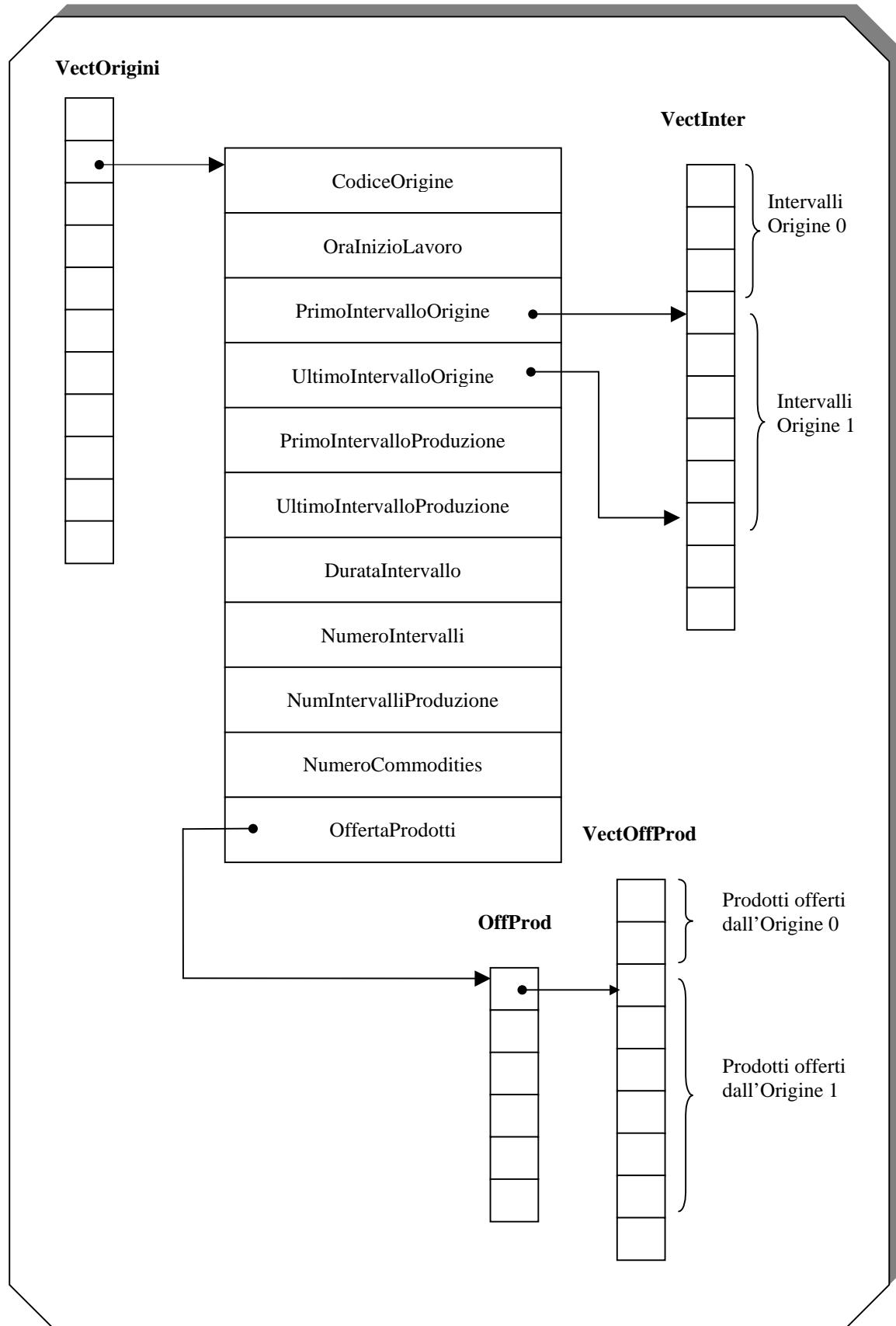
4.2.1 Strutture riguardanti le origini

La struttura **Origine**, schematizzata in Fig.4.1, contiene sia informazioni lette direttamente dal file ori.exp che campi il cui valore è ottenuto attraverso la chiamata di metodi privati, o anche informazioni ottenute leggendo altri files della stessa directory. Per ogni origine i dati vengono memorizzati in strutture di tipo Origine, inoltre nel vettore **VectOrigini** sono memorizzati i puntatori a tali strutture. Il campo **OraInizioLavoro** coincide con l’ora di inizio della giornata lavorativa (espressa in minuti anziché in ore, come nel file di input) dell’origine considerata.

Il vettore **OffProd** è un array di indici che collega la struttura Origine con il vettore di strutture OffertaProdPerOri, che descriveremo in seguito, ma che essenzialmente contiene informazioni riguardanti l’offerta di prodotto per ogni origine, in particolare l’indice dell’origine che lo offre. Tutti gli indici degli elementi che nel vettore VectOffProd si riferiscono all’origine i-esima sono contenuti in OffProd dell’origine i .

Per quanto riguarda gli altri campi contenuti in Origine è necessario fare una ulteriore precisazione. Abbiamo visto che, nel modello matematico, il tempo è stato discretizzato suddividendo ogni origine in un intervallo di tempo lungo Δ_i , ossia il tempo medio di caricamento all’origine i.

Vediamo quindi come si ottiene il valore Δ_i e di conseguenza come viene suddivisa l’origine.



[Fig.4.1 :Schema della struttura Origine]

4.2.1.1 Calcolo del tempo medio di caricamento

Questo calcolo è stato effettuato attraverso il metodo privato della classe “ContaProdottiPerOrigine(...)”, il quale restituisce un vettore di due elementi: il primo è il numero di commodities, ossia il numero di prodotti che l’origine considerata effettivamente fornisce, che verrà memorizzato nel campo **NumeroCommodities** della struttura Origine; il secondo è il valore Δ_i , che verrà inserito nel campo **DurataIntervallo**.

La durata media Δ_i è ottenuta nel seguente modo: per l’origine i si leggono dal file oripro.exp i dati riguardanti la produzione giornaliera, lo stock iniziale all’inizio della giornata e la durata di caricamento prevista per ogni prodotto. Se, per un determinato prodotto, sono verificate:

- durata di caricamento < 23.59 (cioè è possibile caricare il prodotto);
- offerta = stock iniziale + produzione giornaliera > 0;
- offerta > capacità di carico ,cioè è possibile caricare almeno un camion (un camion viene considerato carico quando si raggiunge l’80% della sua capacità);

allora quel prodotto va incluso nel numero di commodities (NC) dell’origine i e la sua durata di caricamento entrerà a far parte del calcolo della durata media.

Alla fine del ciclo sui prodotti dell’origine i :

$$\Delta_i = \frac{Durata\ Totale}{NC}$$
 dove *DurataTotale* è la somma delle durate relative ai prodotti che verificano le condizioni suddette.

4.2.1.2 Determinazione degli intervalli temporali all’origine

Si parte dai seguenti dati, letti da file o determinati come descritto nei precedenti paragrafi:

Durata Intervallo = Δ_i ;

Tempo di Inizio della giornata lavorativa = ORIMIN;

Tempo di Fine della giornata lavorativa = ORIEND = ORIMIN + ORINOR + ORISTR ;

Ora Inizio della Produzione = ORIINI;

Ora Fine della Produzione = ORIEND1 (ORIINI + ORIDUR)

e si modifica la giornata lavorativa in modo da renderla multipla della durata dell’intervallo nel seguente modo:

Primo Intervallo = 0

$$\text{Ultimo Intervallo} = \frac{ORIEND - ORIMIN}{\Delta_i} - 1$$

$$\text{Nuovo Tempo di Fine} = (\text{Ultimo Intervallo} + 1) * \Delta_i$$

Anche la durata della giornata di produzione viene modificata in modo da renderla multipla di Δ_i .

$$\text{Primo Intervallo Produzione} = \frac{ORIINI - ORIMIN}{\Delta_i}$$

$$\text{Nuovo Inizio Produzione} = \text{Primo Intervallo Prod} * \Delta_i;$$

$$\text{Ultimo Intervallo Produzione} = \frac{ORIEND1 - ORIINI}{\Delta_i} + \text{Primo Intervallo Prod} - 1;$$

$$\text{Nuova Fine Produzione} = (\text{Ultimo Intervallo Produzione} + 1) * \Delta_i$$

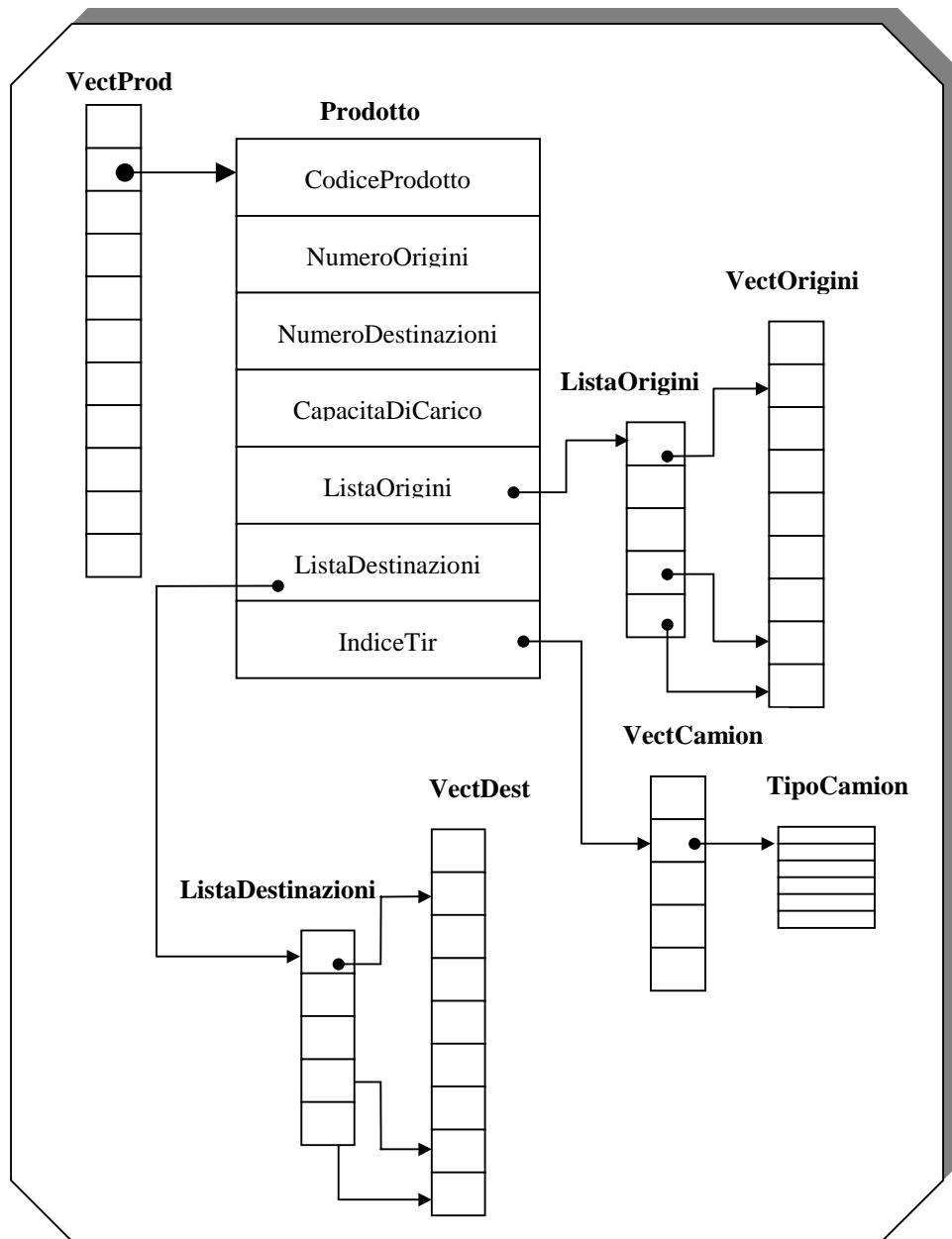
$$\text{Numero Intervalli Produzione} = \text{Ultimo Intervallo Produzione} - \text{Primo Intervallo Produzione} + 1$$

L’Esempio 1, nell’Appendice B illustra il procedimento descritto. Ritornando alla struttura Origine, i campi **PrimoIntervalloProduzione**, **UltimoIntervalloProduzione**, **NumeroIntervalliProduzione** e **NumeroIntervalli** sono calcolati come appena descritto. Per quanto riguarda invece **PrimoIntervalloOrigine** e **UltimoIntervalloOrigine**, la numerazione tiene conto anche della posizione occupata dall’origine all’interno di VectInter, come si vede dallo schema in Fig.4.1.

Andiamo ora ad analizzare la struttura **Prodotto**, schematizzata in Fig.4.2. Tale struttura comprende 7 campi che contengono tutte le informazioni riguardanti i diversi tipi di legno forniti dalle origini. Alcune, come **CapacitaDiCarico** e **CodiceProdotto** sono ottenute rispettivamente dai files tipro.exp e pro.exp. Altre sono calcolate con metodi privati: **NumeroOrigin** è ottenuta con il metodo “OriginiPerProdotto(...)”, che conta il numero delle origini che forniscono un determinato prodotto; analogamente “DestinazioniPerProdotto(...)” fornisce il **NumeroDestinazioni** che ne fanno richiesta. Questi due dati sono necessari anche per dimensionare i vettori **ListaOrigin** e **ListaDestinazioni** che contengono rispettivamente gli indici delle origini [destinazioni] all’interno del vettore VectOri [VectDest] che offrono [richiedono] il prodotto. Il campo **IndiceTir** contiene, infine l’indice dell’unico tipo di camion (nel vettore VectCamion, che descriveremo in seguito) in grado di trasportare il prodotto.

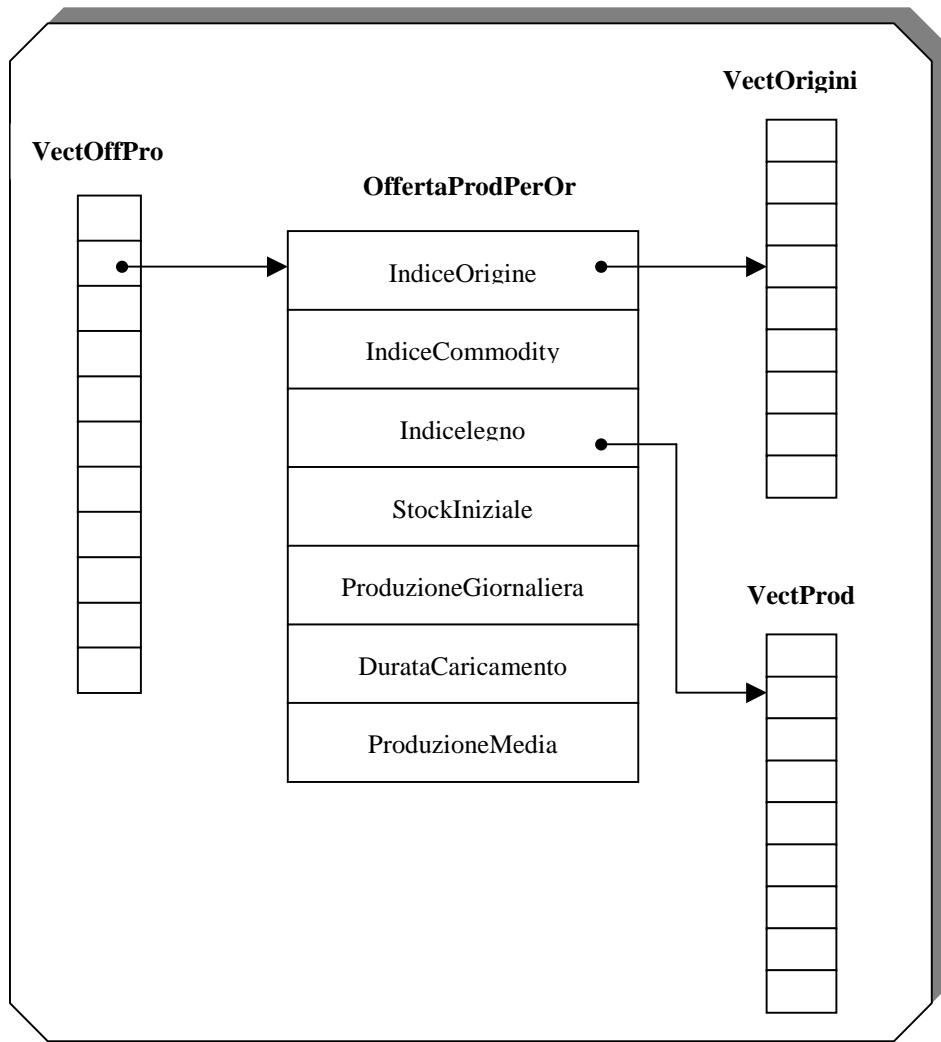
A proposito di quest’ultimo dato è necessario fare una precisazione. Abbiamo visto, nel Cap.2, che il modello matematico impone che ogni prodotto possa essere trasportato da un solo tipo di camion. Questa restrizione nelle istanze non è sempre rispettata: talvolta un prodotto può essere trasportato con diversi tipi di camion. E’ stato quindi necessario, per rispettare il modello, fare una scelta e associare univocamente un determinato prodotto ad uno specifico tipo di camion. Poiché non sussistono, in generale, condizioni di dominanza tra i vari tipi di camion con i quali è possibile trasportare un prodotto (cioè non è possibile stabilire il camion più economico rispetto a tutti i

parametri di costo), la scelta adottata è stata quella di assegnare semplicemente al prodotto il primo camion presente nell'elenco dei mezzi utilizzabili per quello specifico prodotto.



[Fig.4.2: Schema della struttura Prodotto]

Nel descrivere la struttura Origine si è parlato della struttura **OffertaProdPerOri** alla quale è collegata. Come si può vedere nello schema in Fig.4.2 sono qui contenute le informazioni che si riferiscono ad ogni specifica offerta di prodotto.



[Fig. 4.3: Schema della struttura OffertaProdPerOri]

Il generico elemento del vettore **VectOffProd** viene creato in base alla lettura del file oripro.exp e se sono verificate le condizioni del paragrafo 4.2.1.1. In questo caso **IndiceOrigine** contiene l'indice dell'origine, nel vettore VectOrigini, alla quale si riferisce tale offerta; **IndiceLegno** l'indice del prodotto, in VectProd, che viene offerto; **IndiceCommodity** indica, tra le commodity associate all'origine, se si tratta del primo prodotto offerto, oppure del secondo, e così via. **StockIniziale**, **ProduzioneGiornaliera** e **DurataCaricamento** sono lette dal file di input e l'unica operazione effettuata è stata quella di trasformare le unità di misura; in particolare, i dati in metri cubi sono stati trasformati i decimetri cubi, mentre il tempo espresso in ore è stato convertito in minuti. Questa operazione non è locale a questi dati, ma si è resa necessaria, per esigenze di programmazione, con tutti i dati dello stesso genere. Il vantaggio di tale scelta è quello di poter

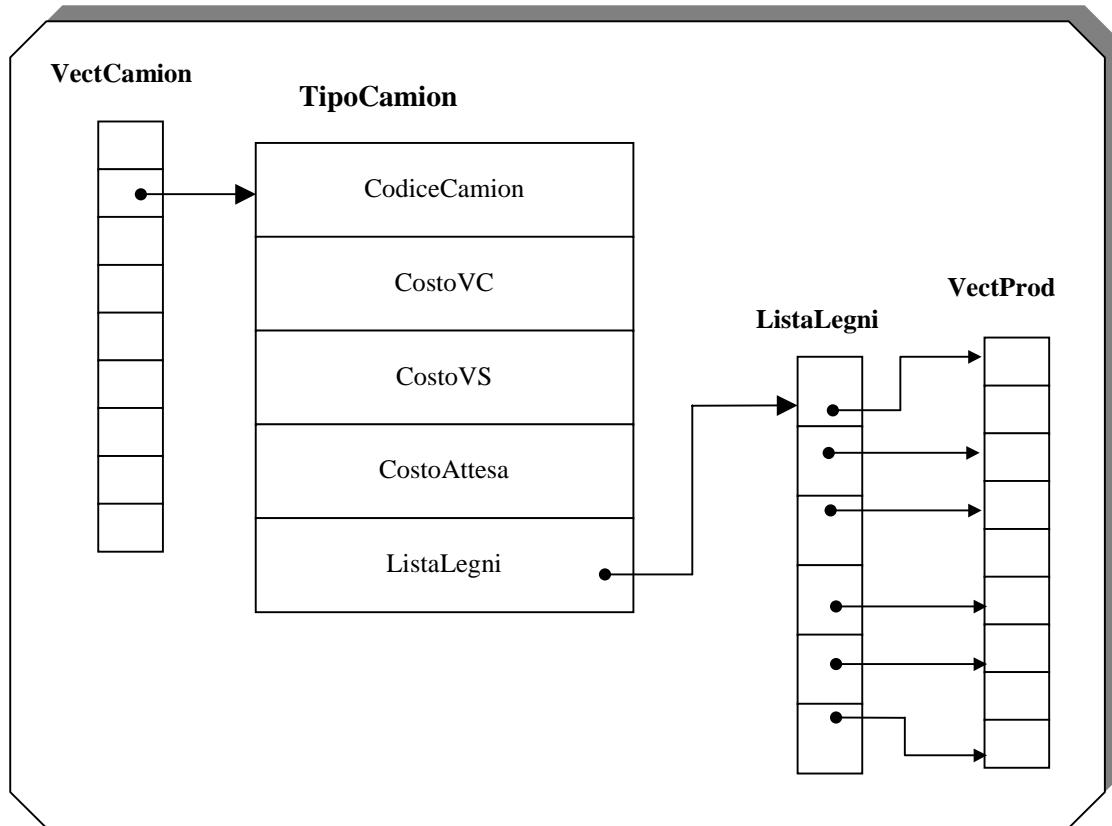
memorizzare tutti i valori come numeri interi: oltre a consentire un risparmio in memoria, ciò evita gli errori di arrotondamento nei calcoli che altrimenti potrebbero creare qualche problema.

Per quanto riguarda **ProduzioneMedia**, il campo viene calcolato come segue:

Produzione_K_I : prodotto K fornito dall'origine I nell'arco della giornata di produzione;
 N_I_P : Numero intervalli produzione (calcolo effettuato nel par. 3.2.2);

$$\text{ProduzioneMedia} = \frac{\text{Produzione_K_I}}{N_I_P}.$$

Andiamo ora descrivere la struttura **TipoCamion**, rappresentata in Fig.4.4, contenente tutte le informazioni associate ai camion. Questa struttura, tra le altre cose, contiene i costi di viaggio carico e scarico (**CostoVC**, **CostoVS**) dei camion.



[Fig.4.4: Schema della struttura TipoCamion]

Un ulteriore spesa associata al viaggio è quella riguardante l'attesa eventuale a destinazione prima che il camion venga scaricato, oppure all'origine prima che venga caricato. Poiché questo costo varia a seconda del tipo di camion, l'informazione relativa è memorizzata nel campo **CostoAttesa**. Infine, dato che ogni tipo di camion è in grado di trasportare diversi tipi di prodotto, nell'array **ListaLegni** sono contenuti gli indici dei prodotti trasportabili (come posizioni in VectProd).

L’ultima struttura, che in qualche modo possiamo considerare attinente con le origini dei viaggi è **InterOri**, schematizzata in Fig.4.5. Sono qui mantenute tutte le informazioni che si riferiscono ad ogni specifico intervallo temporale in cui le origini sono state suddivise. In particolare abbiamo **OriIntervallo** che ci indica a quale origine, nel vettore VectOrigini, l’intervallo considerato si riferisce. Per quanto riguarda il campo **NumInterOri** bisogna ricordare che nella struttura Origini i dati PrimoIntervallo e UltimoIntervallo non si riferivano ai valori effettivamente calcolati nella sezione 4.2.1.2, ma erano le posizioni all’interno del vettore VectInter di strutture InterOri. Pertanto ora memorizziamo, proprio attraverso NumInterOri la numerazione effettiva dell’intervallo, ossia se si tratta del secondo intervallo, o del terzo e così via.

4.2.1.3 Calcolo dei deficit relativi ad ogni intervallo dell’origine

Vediamo adesso come vengono calcolate le quantità:

- Stock_K_I : stock di prodotto K presente all’inizio della giornata all’Origine I;
 Produzione_K_I : prodotto K fornito dall’origine I nell’arco della giornata di produzione;
 CapacitaTir : capacità di caricamento del camion che trasporta il prodotto K;
 P_Int_K_I : produzione media relativa ad ogni intervallo di produzione;
 Avanzo_Prodotto_K_I_inter_J : quantità residua di prodotto K al termine dell’intervallo J;
 Offerta_Prod_K_I_inter_J : offerta di prodotto K relativa all’intervallo J, espressa in numero di camion che si possono caricare;
 Deficit_K_I_inter_J : deficit all’intervallo J dell’origine I per la commodity K;

Il procedimento che andremo ora a descrivere riguarda l’Origine I e il prodotto K. Naturalmente questo procedimento, di cui viene presentato un esempio nell’Appendice B, va effettuato per tutte le origini, e all’interno di ognuna di queste per tutte le commodity.

CALCOLO DEL DEFICIT RELATIVO ALL’INTERVALLO 0 (primo nodo dell’Origine I)

Se Primo Intervallo Produzione = 0
 ossia gli intervalli di produzione coincidono con quelli di lavoro

$$\text{Deficit_K_I_inter_0} = \frac{(\text{Stock_K_I} + \text{P_Int_K_I})}{\text{CapacitaTir}}$$

$$\text{Avanzo_Prodotto_K_I_inter_0} = (\text{Stock_K_I} + \text{Produzione_Inter_K_I}) - (\text{Deficit_K_I_inter_0} * \text{CapacitaTir})$$

Se Primo Intervallo Produzione $\neq 0$
 ossia gli intervalli di produzione non coincidono con quelli di lavoro

$$\text{Deficit_K_I_inter_0} = \frac{(\text{Stock_K_I})}{\text{CapacitaTir}}$$

$$\text{Avanzo_Prodotto_K_I_inter_0} = (\text{Stock_K_I}) - (\text{Deficit_K_I_inter_0} * \text{CapacitaTir})$$

CALCOLO DEL DEFICIT RELATIVO AGLI ALTRI INTERVALLI

Sia J l'indice dell'intervallo che stiamo considerando:

Prima fase:

- Se $J \in \{ \text{Primo Intervallo Produzione, Ultimo Intervallo Produzione} \}$

$$\text{Avanzo_Prodotto}_K_I_{\text{inter}_J} = \text{Avanzo_Prodotto}_K_I_{\text{inter}_J-1} + \text{Produzione}_I$$

Seconda fase:

- Se $\text{Avanzo_Prodotto}_K_I_{\text{inter}_J} \geq \text{CapacitaTir}$ (ossia è possibile riempire un camion)

$$\text{Offerta_Prod}_K_I_{\text{inter}_J} = \frac{\text{Avanzo_Prodotto}_K_I_{\text{inter}_J}}{\text{CapacitaTir}}$$

$$\text{Aggiunta}_I = \text{Aggiunta}_I_{-1} + \text{Offerta_Prod}_K_I_{\text{inter}_J}$$

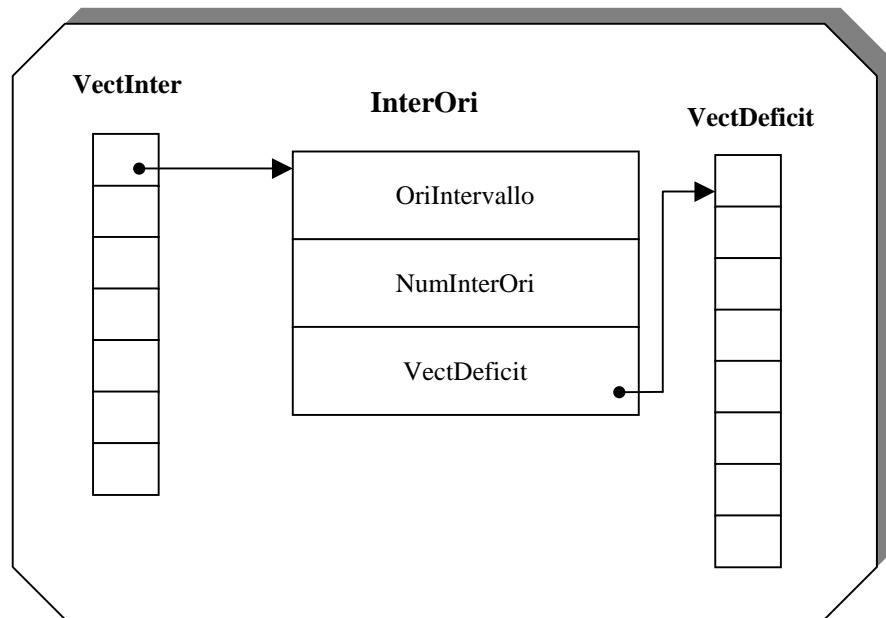
$$\text{Avanzo_Prodotto}_K_I_{\text{inter}_J} = \text{Avanzo_Prodotto}_K_I_{\text{inter}_J-1} - (\text{Offerta_Prod}_K_I_{\text{inter}_J} * \text{CapacitaTir})$$

- Se $\text{Avanzo_Prodotto}_K_I_{\text{inter}_J} < \text{CapacitaTir}$ (ossia non è possibile riempire un camion)

$$\text{Offerta_Prod}_K_I_{\text{inter}_J} = 0$$

In ogni caso al termine della seconda fase si ha:

$$\text{Deficit}_K_I_{\text{inter}_J} = \text{Offerta_Prod}_K_I_{\text{inter}_J}$$

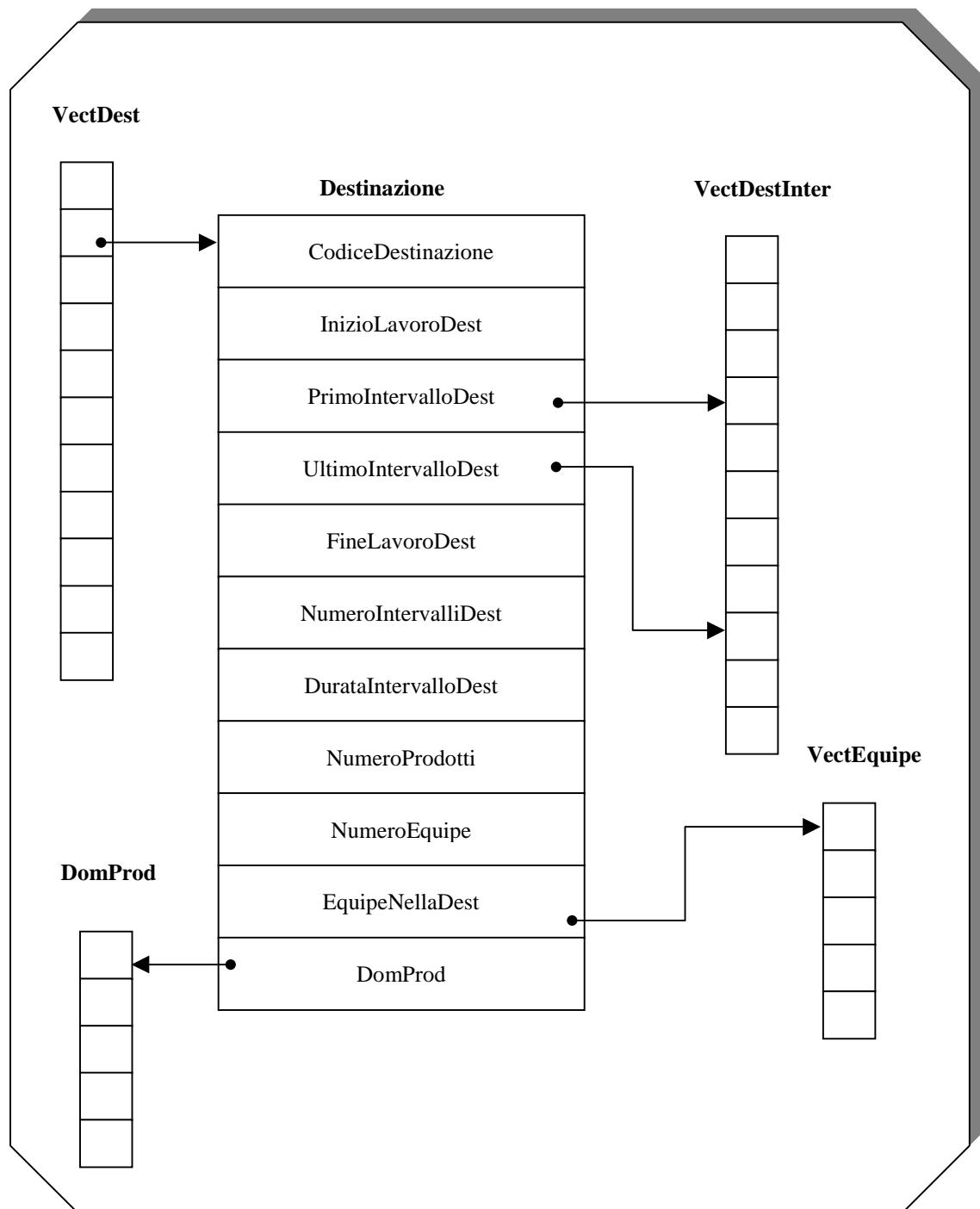


[Fig.4.5: Schema della struttura InterOri]

4.2.2 Strutture riguardanti le destinazioni

Innanzitutto, analogamente ad Origini abbiamo la struttura **Destinazione**, raffigurata in Fig.4.6, che contiene informazioni dello stesso genere. Anche a destinazione, infatti, la giornata è stata

suddivisa in intervalli temporali. Il procedimento è più complesso in quanto bisogna tenere conto del fatto che a destinazione ci sono a disposizione diverse attrezzature per lo scaricamento del legname, e queste attrezzature non sono disponibili tutti negli stessi orari.



[Fig.4.6: Schema della struttura Destinazione]

Teniamo quindi in **NumeroEquipe** il numero delle equipe che lavorano alla Destnazione cosiderata e in **EquipeNellaDest** gli indici di tali equipe, come posizioni all'interno del vettore **VectEquipe**.

4.2.2.1 Determinazione degli intervalli temporali a destinazione

Innanzitutto è necessario calcolare la durata della giornata lavorativa a destinazione tenendo conto delle diverse equipe di scaricamento (i dati necessari sono contenuti nel file equ.exp).Dati:

Ora Inizio Giornata Lavorativa Equipe X = EQUINI

Ora Fine Giornata Lavorativa Equipe X = EQUINI + EQUDUR (EQUEND)

Il tempo di inizio e di fine della giornata sono ottenuti come segue:

Ora Inizio Destinazione I:	il più piccolo orario di inizio delle equipe che lavorano alla destinazione I e possono scaricare i prodotti richiesti = MIN
Ora Fine Destinazione I:	il più grande orario di fine lavoro delle equipe che lavorano alla destinazione I e possono scaricare i prodotti richiesti = MAX

Queste informazioni sono memorizzate nei campi **InizioLavoroDest** e **FineLavoroDest** della destinazione relativa. La giornata così calcolata è divisa in intervalli temporali. La durata dell'intervallo è il Massimo Comun Divisore (MCD) calcolato tra i diversi tempi di scaricamento delle equipe, come si può veder dall'Esempio 3 nell'Appendice B. Tale durata è memorizzata in **DurataIntervalloDest**.

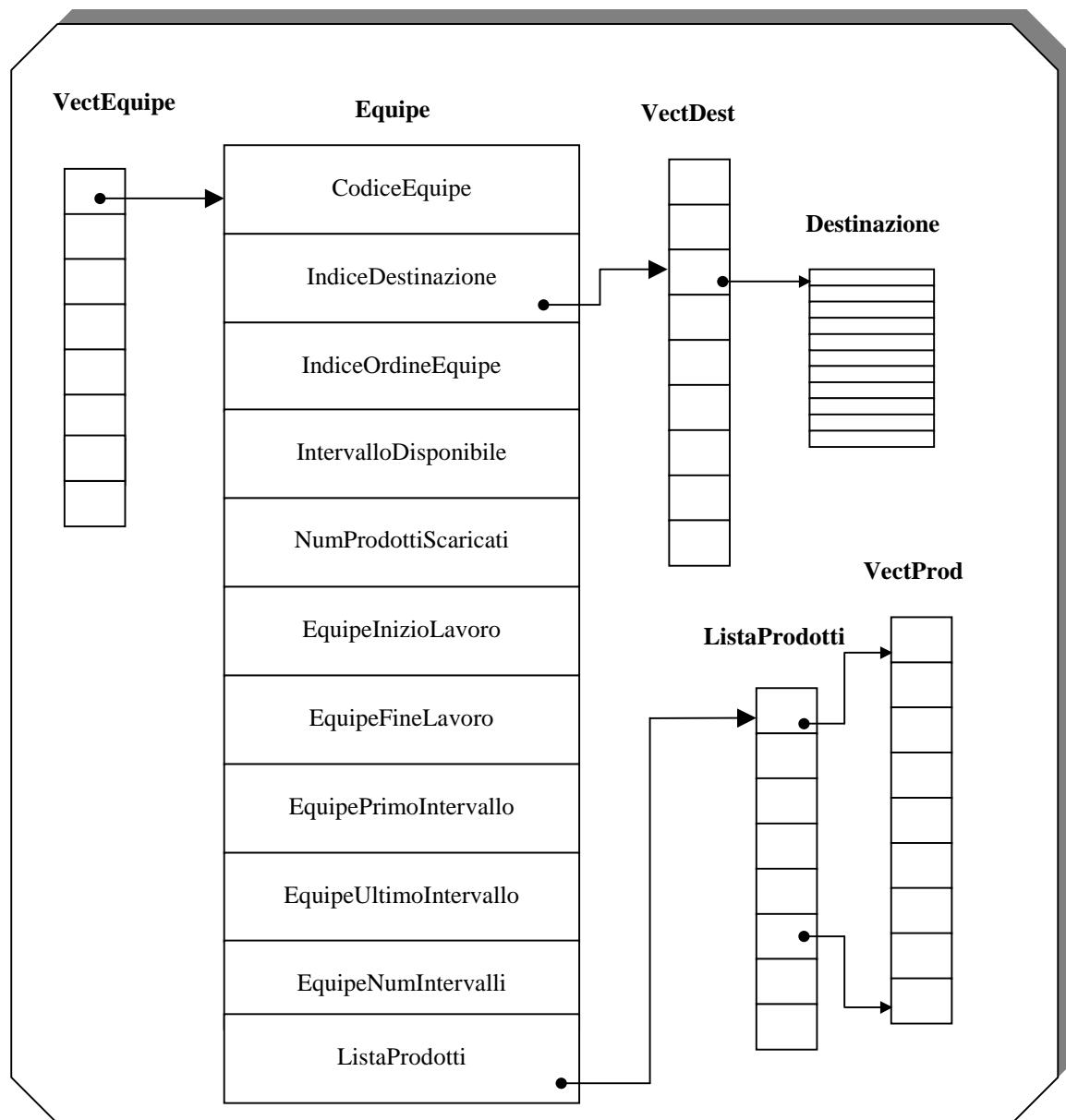
Si noti che il calcolo degli intervalli è diverso tra origine e destinazione: nel primo caso si considera la media, mentre nel secondo si considera il MCD. Il motivo è il seguente: alle destinazioni la domanda è assegnata direttamente, quindi è noto esattamente quanti e quali intervalli. Alle origini, invece, non è noto a priori quale intervallo sarà usato per caricare i prodotti. Questo verrà deciso dal modello matematico. Pertanto viene utilizzata la media come stima del tempo necessario per caricare ogni prodotto.

A questo punto si modifica la giornata lavorativa a destinazione per renderla multipla della durata dell'intervallo.

$$\begin{aligned}
 \text{Primo Intervallo} &= 0 \\
 \text{Nuovo Tempo di Inizio} &= \text{MIN} \\
 \text{Ultimo Intervallo} &= \frac{\text{MAX} - \text{MIN}}{\text{MCD}} - 1 \\
 \text{Nuovo Tempo di Fine} &= (\text{Ultimo Intervallo} + 1) * \text{MCD}
 \end{aligned}$$

Questi valori non sono memorizzate nei campi **PrimoIntervalloDest** e **UltimoIntervalloDest** così come li abbiamo calcolati, bensì all'interno della struttura Destinazione si tiene conto della posizione che tali intervalli occupano nel vettore VectDestInter, che descriveremo in seguito. L'ultima fondamentale informazione è quella contenuta in **NumeroProdotti** e **DomProd** : qui, infatti, abbiamo il numero totale dei prodotti richiesti e i loro indici all'interno del vettore

VectDomProd. Anche le informazioni sulle diverse equipe che lavorano a destinazione devono essere memorizzate, pertanto abbiamo anche la struttura **Equipe**, schematizzata in Fig.4.7.



[Fig.4.7: Schema della struttura Equipe]

4.2.2.2 Modifica della giornata di lavoro delle equipe

Anche la durata della giornata di lavoro delle equipe si modifica in modo da renderla multipla della durata dell'intervallo. Per ogni equipe il procedimento è il seguente:

$$\text{Equipe Primo Intervallo} = \frac{EQUINI - MIN}{MCD}$$

Equipe Nuovo Tempo di Inizio = MIN

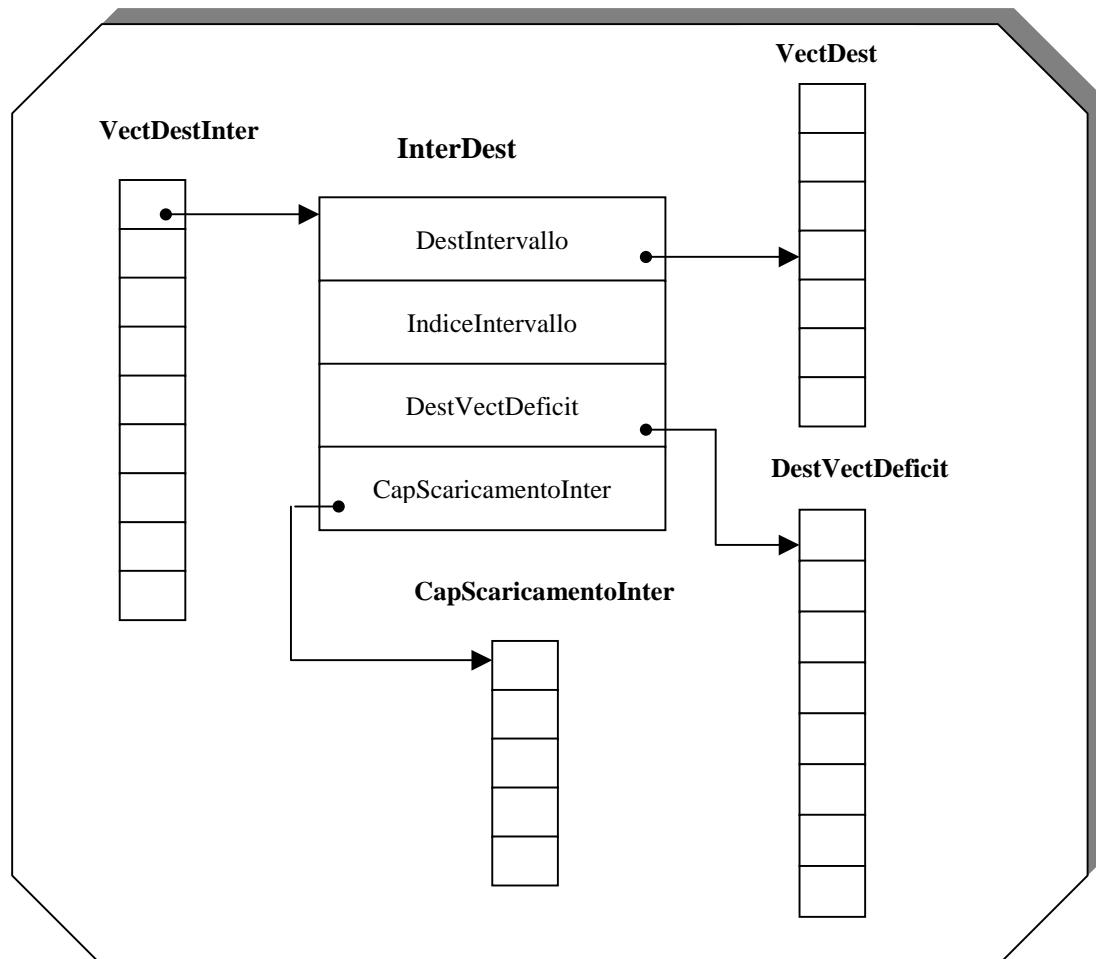
Ultimo Intervallo Equipe = Equipe Primo Intervallo * MCD

Equipe Nuovo Tempo di Fine = (Ultimo Intervallo Equipe + 1) * MCD

Numero Intervalli Equipe = Ultimo Intervallo Equipe - Equipe Primo Intervallo + 1

A questo punto bisogna calcolare la capacità di scarico di ogni intervallo: in pratica si andranno a sommare le attrezzature disponibili negli intervalli considerati.

Così come per le origini abbiamo la struttura **InterOri** adesso per le destinazioni abbiamo la struttura **InterDest** (schematizzata in Fig.4.8) in cui sono inserite le informazioni riguardanti la disponibilità delle equipe e la domanda dei prodotti.



[Fig.4.8: Schema della struttura InterDest]

Poiché all'interno di Destinazione la numerazione degli intervalli era relativa alla posizione in VectInterDest in **IndiceIntervallo** teniamo l'indice dell'intervallo che si sta considerando all'interno di una data destinazione, ed anche la destinazione a cui ci riferiamo, come posizione in VectDest (in **DestIntervallo**). Il vettore **CapScaricamentoInter** ha dimensione pari al numero di equipe che sono disponibili nella destinazione a cui l'intervallo si riferisce. In corrispondenza di ogni equipe, il vettore contiene 1 o 0 a seconda che l'equipe sia disponibile o meno nell'intervallo. L'Esempio 4 nell'Appendice B chiarisce come si determina tale disponibilità.

4.2.2.3 Calcolo del primo e ultimo intervallo di distribuzione

Prima di procedere alla descrizione dell'euristica, ossia delle scelte fatte per distribuire la domanda di prodotto tra gli intervalli temporali in cui è stata suddivisa la giornata lavorativa della destinazione, andremo ad analizzare il modo in cui è stato calcolato, per ogni prodotto

- Primo_Intervallo_Distribuzione: primo intervallo a partire dal quale sarà possibile ricevere a destinazione la prima unità di prodotto richiesta;
- Ultimo_Intervallo_Distribuzione: ultimo intervallo entro il quale potrà essere ricevuta l'ultima unità dello stesso prodotto.

Il procedimento è il seguente:

```

{forall destinazione j ( che richiede almeno un prodotto )
  {forall prodotto k richiesto da j
    {forall origine i ( che offre almeno un prodotto )
      Se è possibile il viaggio tra i e j
        allora
          Ultimo_Intervallo = ultimo intervallo a destinazione in cui si può scaricare l'ultima unità;
          Ultimo_Tempo_Offerta = CalcolaUltimoTempoOfferta_Per_j_da_i(Ultimo_Intervallo );
          Se i può offrire il prodotto k a j /* cioè se i offre il prodotto k ed esiste Ultimo_Tempo_Offerta */
            PrimoTempoDiOfferta_K_I = CalcolaPrimoTempoDiOfferta( i,k );
            UltimoTempoRicezione = TempoViaggio + Ultimo_Tempo_Offerta;
            PrimoTempoRicezione = TempoViaggio + PrimoTempoDiOfferta_K_I;
            Se ( ( PrimoTempoRicezione ≥ Ora_Inizio_Lavoro_Destinazione )
              ∨ ( ( Ora_Inizio_Lavoro_Destinazione – PrimoTempoRicezione ) ≤ MaxAttesa ) ) )
              {
                Min_Tempo_Prima_Ricezione = min(Min_Tempo_Prima_Ricezione,
                                              PrimoTempoRicezione );
                Max_Tempo_Ultima_Ricezione = max(Max_Tempo_Ultima_Ricezione,
                                              UltimoTempoRicezione);
              }
            }
          Primo_Intervallo_Distribuzione_K = Intervallo a destinazione che include Min_Tempo_Prima_Ricezione;
          Ultimo_Intervallo_Distribuzione_K = Intervallo a destinazione che include Max_Tempo_Ultima_Ricezione}
    }
  }
}

```

Vediamo più in dettaglio i passi della procedura:

- **Ultimo_Intervallo** : è l'ultimo intervallo della destinazione entro il quale è possibile scaricare l'ultima unità di prodotto richiesto (questo vale per ogni prodotto domandato); viene calcolato nel seguente modo:

$$\forall \text{ prodotto K}$$

$$\text{Ultimo_Intervallo} = 0;$$

$$\forall (\text{ (Equipe Q che lavora nella destinazione J) } \wedge \text{ (Q scarica il prodotto K) })$$

$$\text{Ultimo_Intervallo} = \max (\text{Ultimo_Intervallo}, \text{Ultimo intervallo di lavoro dell'equipe Q})$$

- **PrimoTempoDiOfferta_K_I** : è il primo tempo di offerta del prodotto k da parte dell'origine I, cioè è il tempo in cui finisce il primo intervallo a partire dal quale l'origine è pronta ad offrire la prima unità di prodotto K. Se l'origine ha immagazzinato dello stock dalla giornata precedente in quantità sufficiente a formare una “camionata” di prodotto, allora il

$$\text{PrimoTempoDiOfferta_K_I} = \text{Inizio Giornata Lavorativa_I} + \text{Durata Intervallo Origine_i}$$

In questo calcolo si prende la fine dell'intervallo e non l'inizio perché va considerato il tempo necessario per il caricamento del prodotto (ossia la durata stessa dell'intervallo) ed è a partire da questo momento che un'unità di prodotto K è disponibile all'origine I.

- **Ultimo_Tempo_Offerta** : all'origine non esiste un ultimo tempo di caricamento (ossia un tempo oltre il quale non è più possibile spedire prodotto), quindi, come Ultimo_Tempo_Offerta di un qualsiasi prodotto K da parte dell'origine I per la destinazione J, viene preso il tempo di fine dell'ultimo intervallo di I a partire dal quale (considerando il tempo di viaggio tra I e J) è possibile arrivare a destinazione entro l'inizio di Ultimo_Intervallo o, al limite, in un tempo che rientri nell'attesa massima consentita prima dello scaricamento. La procedura che calcola Ultimo_Tempo_Offerta restituisce “-1” se non trova un intervallo dell'origine compatibile con Ultimo_Intervallo. A questa procedura vengono passati: l'indice dell'origine, l'indice della destinazione, Ultimo_Intervallo e tempo di viaggio tra origine e destinazione.

```

{
  Y = Indice dell'ultimo intervallo dell'origine i;
  TROVATO = False;
  Ora_Inizio_Ultimo_Intervallo = ora in cui inizia Ultimo_Intervallo a destinazione
  Ora_Fine_Y = orario di fine dell'intervallo Y;
  Finchè (( Y ≥ Indice_Primo_Intervallo_Origine_I ) ∧ ( ¬ TROVATO ))
  {
    ...
  }
}
  
```

```

Fine_Viaggio = Ora_Fine_Y + TempoViaggioCarico(I,J);
Tempo_Atesa = Ora_Inizio_Ultimo_Intervallo – Fine_Viaggio;

```

```

Se ( 0 ≤ Tempo_Atesa ≤ Max_Atesa)      /* Condizione 1*/
    TROVATO = True;
Se ( Tempo_Atesa > 0 ∧ Tempo_Atesa > Max_Atesa ) /* Condizione
    TROVATO = True;
Se ( (non vale Condizione1) ∧ (non vale Condizione 2) )

    Decrementa l'indice Y;
    Ora_Fine_Y = Ora_Fine_Y – Durata dell'intervallo dell'origine I;
}
Se TROVATO = True
    Ultmo_Tempo_Offerta = Tempo di fine dell'intervallo Y;
altrimenti restituisci(-1);
}

```

- **PrimoTempoRicezione** e **UltimoTempoRicezione** : una volta calcolati il primo e l'ultimo tempo di offerta, aggiungendo il tempo di viaggio, si calcolano il primo e ultimo tempo di ricezione a destinazione ossia il primo tempo a partire dal quale sarebbe possibile ricevere la prima unità di prodotto K e l'ultimo tempo entro il quale sarebbe possibile ricevere l'ultima unità del medesimo prodotto, dall'origine I.
- **Primo_Intervallo_Distribuzione_K** e **Ultimo_Intervallo_Distribuzione_K**: calcolati rispettivamente prendendo il più piccolo tra tutti i tempi di prima ricezione (Min_Tempo_Prima_Ricezione) e il più grande tra tutti i tempi di ultima ricezione (Max_Tempo_Ultima_Ricezione), si calcolano, infine, gli intervalli, a destinazione, in cui essi ricadono.

3.2.8 Calcolo dei deficit relativi ad ogni intervallo della destinazione

A questo punto è possibile vedere in che modo si è scelto di “partizionare”, negli intervalli compresi tra il primo e l'ultimo di distribuzione, calcolati nella sezione precedente, la domanda complessiva dei prodotti richiesti dalla destinazione. A questo scopo sono state implementate euristiche diverse che hanno portato a diverse conclusioni. La prima prevede una distribuzione, delle unità richieste, a partire dal primo intervallo utile: ha portato ad una concentrazione degli scaricamenti nella prima parte della giornata lavorativa ed ha comportato l'inammissibilità di alcune istanze a causa dell'impossibilità delle origini di rifornire le destinazioni prima che sia stata

raggiunta una certa produzione. Per “ricadere” nell’ammisibilità, si è fatto in modo che il lavoro delle unità di scaricamento venisse spostato verso le ore centrali della giornata lavorativa. Questo è stato ottenuto attraverso l’introduzione di due parametri: x e δ , il primo avente lo scopo di “rallentare”, nello scaricamento, le destinazioni (che risultano essere più veloci di quanto lo siano le origini nel produrre); il secondo avente lo scopo di “centralizzare” l’attività lavorativa delle attrezzature di scaricamento. Svantaggio di questa scelta è la “perdita” di unità di prodotto richieste: questo succede proprio a causa della diminuzione degli intervalli in cui distribuire.

La seconda euristica non ha questo inconveniente, ma può in alcuni casi portare a costi più alti. Vediamo innanzitutto in cosa consiste: anziché distribuire dal primo intervallo in poi, si inizia a distribuire dall’ultimo a ritroso fini al primo. Il motivo è dettato proprio dalla necessità di consentire alle origini di produrre il più possibile prima che le destinazioni inizino a richiedere.

Si è detto che in questa scelta i costi potrebbero essere più elevati, il motivo è il seguente: per consentire ad alcune origini di rifornire alcune destinazioni, è necessario fissare il parametro d’attesa abbastanza alto, di conseguenza ciò che spendiamo in più è dato proprio dal costo di questo “tempo morto”. Il procedimento è il seguente:

```

{ ∀ destinazioneJ
  { ∀ prodotto domandato K
    Se la quantità domandata da K è > 0
      Trova_Intervalli_Per_Distribuire
    }
  }
}

```

- Trova_Intervalli_Per_Distribuire: è una procedura che utilizza il metodo “SelezioneEquipe()”, il quale controlla se tra gli intervalli in cui sono ancora a disposizione delle attrezzature per lo scaricamento ce n’è un numero consecutivo sufficiente per lo scaricamento del prodotto in questione. Poiché ogni prodotto può essere scaricato da una o più équipe che lavorano nella destinazione considerata, vedremo, nella descrizione della struttura DomProdPerDest, che i loro indici sono memorizzati nel vettore ListaEquipe; pertanto la ricerca di una équipe che possa scaricare il prodotto sarà limitata alle sole équipe contenute il tale vettore. La procedura “SelezioneEquipe(..)” è la seguente:

```

TROVATA = False;
j = 0;
Finchè ( ( j > Numero delle Equipe che possono scaricare il prodotto ) ∧
          ( ¬ TROVATA ) )

```

{

NumInter = numero degli intervalli necessari all'equipe per scaricare il prodotto;

Se (Numero intervalli in cui l'equipe è disponibile) ≥ NumInter

{

Trova_Intervalli_Consecutivi (Numero_Equipe, Inizio, NumInter,
Fine, Risultati);

Se (Risultati.Trovato)

TROVATA = True;

altrimenti j = j + 1;

}

}

I parametri passati alla procedura **Trova_Intervalli_Consecutivi** hanno il seguente significato:

- **Numero_Equipe** : informazione presa dal campo IndiceOrdineEquipe della struttura Equipe;
- Inizio: indice dell'intervallo da cui iniziare la ricerca nel vettore degli intervalli, più precisamente:

Inizio = max (Primo intervallo di lavoro dell'equipe,
Primo intervallo di distribuzione del prodotto)

- Fine = indice dell'ultimo intervallo, in VectInter, al quale terminare la ricerca, più precisamente:

Fine = min (Ultimo intervallo di lavoro dell'equipe,
Ultimo intervallo di distribuzione del prodotto)

- **NumInter** : numero di intervalli necessari all'equipe per scaricare un camion di prodotto;
- **Risultato** : è la struttura in cui saranno restituiti i risultati. È composta da due campi: il primo di tipo booleano che indicherà se la ricerca ha avuto successo, il secondo di tipo Indice in cui sarà memorizzato l'indice dell'intervallo in cui sarà possibile ricevere un camion di prodotto.

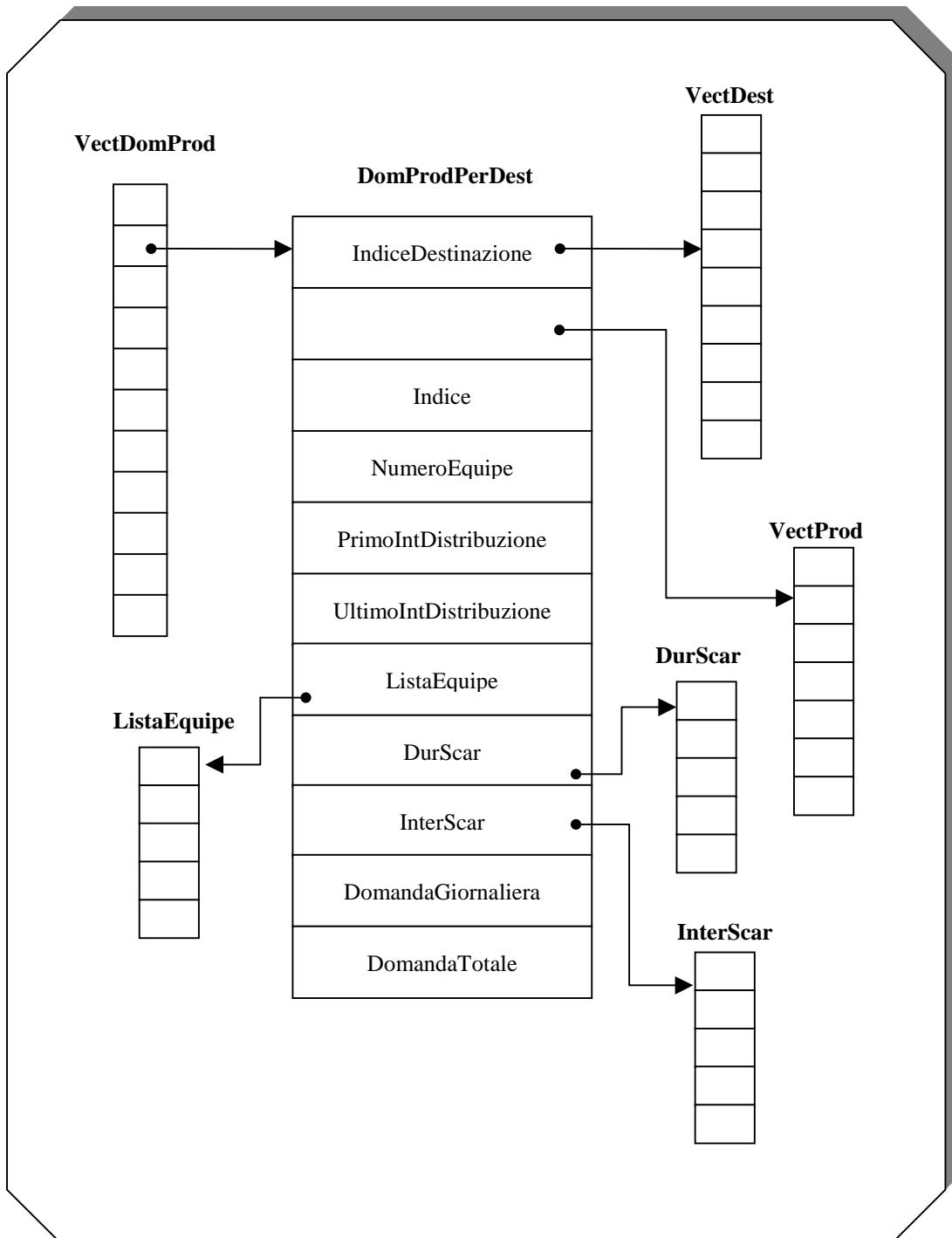
Questa procedura, a seconda dell'euristica usata, inizierà la sua ricerca, nel vettore degli intervalli, a partire dal valore del parametro Inizio fino al valore del parametro Fine, o viceversa a partire da Fine andando a ritroso fino ad arrivare a inizio, e quello che fa non è altro che cercare un numero di intervalli consecutivi sufficienti per scaricare il prodotto con capacità di scarico maggiore di zero.

Se l'equipe viene trovata si andranno ad aggiornare:

- le capacità di scaricamento degli intervalli scelti;
- il numero complessivo di intervalli di disponibilità delle equipe.

Inoltre, al primo di tali intervalli verrà assegnata domanda 1, ossia nel vettore **DestVectDeficit** (nella struttura **InterDest**) in corrispondenza del prodotto distribuito verrà memorizzato il valore “1”.

L’ultima struttura che andiamo a descrivere è **DomProdPerDest**, schematizzata in Fig.4.8, nella quale sono contenute informazioni sulla domanda di ogni prodotto che risulta alle diverse destinazioni.



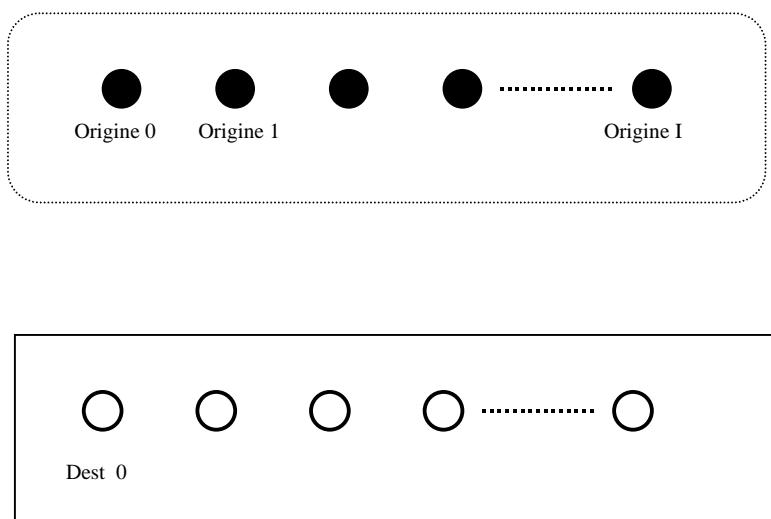
[Fig.4.8:Schema della struttura DomProdPerDest]

I campi **IndiceDestinazione** e **IndiceProdotto** contengono rispettivamente l'indice della destinazione richiedente e l'indice del prodotto oggetto della richiesta; il campo Indice, invece è necessario per sapere l'indice relativo, all'interno della destinazione IndiceDestinazione, del prodotto in questione. **NumEquipe** è il numero delle equipe che possono scaricare IndiceProdotto, inoltre gli indici di queste equipe sono contenuti, come abbiamo visto in 3.2.8, in **ListaEquipe**. Ancora associata alle equipe abbiamo le informazione **DurScar** e **InterScar** che, per ogni indice di equipe contenuto in **ListaEquipe**, contengono rispettivamente le durate di scaricamento, e il numero di intervalli necessari per lo scaricamento. **PrimoIntervalloDistribuzione** e **UltimoIntervalloDistribuzione** sono ottenuti, per ogni domanda di prodotto, come visto in 3.2.7. Infine **DomandaGiornaliera** e **DomandaTotale** contengono rispettivamente la quantità giornaliera di prodotto richiesto (espressa in numero di camion) e la quantità totale di prodotto distribuito (quindi $\text{DomandaTotale} \leq \text{DomandaGiornaliera}$)

3.3 Interfaccia per la classe Transportation

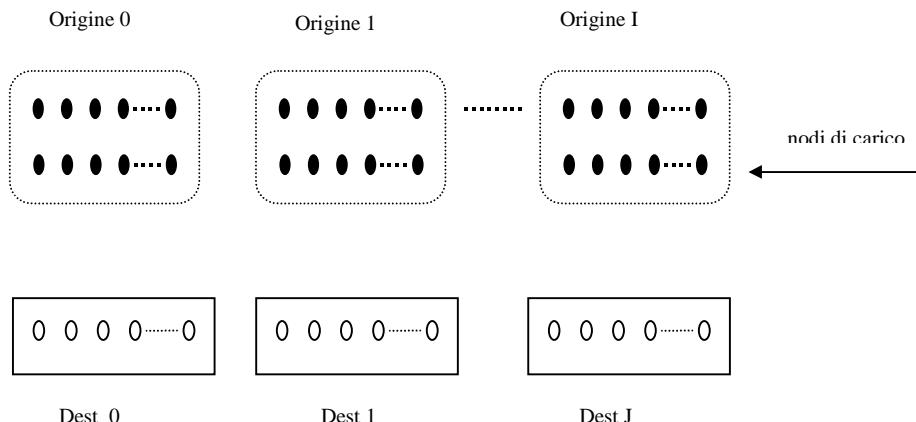
Sulla base delle strutture finora descritte, in particolare il vettore degli intervalli delle origini con le rispettive offerte, ed il vettore degli intervalli delle destinazioni con le rispettive domande, si può passare alla costruzione delle strutture dati che definiscono il grafo “spazio-tempo” visto nel Capitolo 2. Attraverso i metodi pubblici della classe **LetturaDati** queste strutture potranno essere recuperate dalla classe **Transportation** (a cui viene passato un oggetto di tipo **LetturaDati**) che le userà per costruire l'oggetto di tipo **Graph** da passare come al costruttore della classe **MMCFcplex**, interfaccia per il solutore CPLEX. Le classi sopra citate saranno descritte nei prossimi capitoli.

Prima di tutto vediamo in che modo, a partire dalle origini e dalle destinazioni contenute in **VectOrigini** e in **VectDest** siamo arrivati al grafo “spazio-tempo”. La situazione iniziale è quella schematizzata in Fig._.



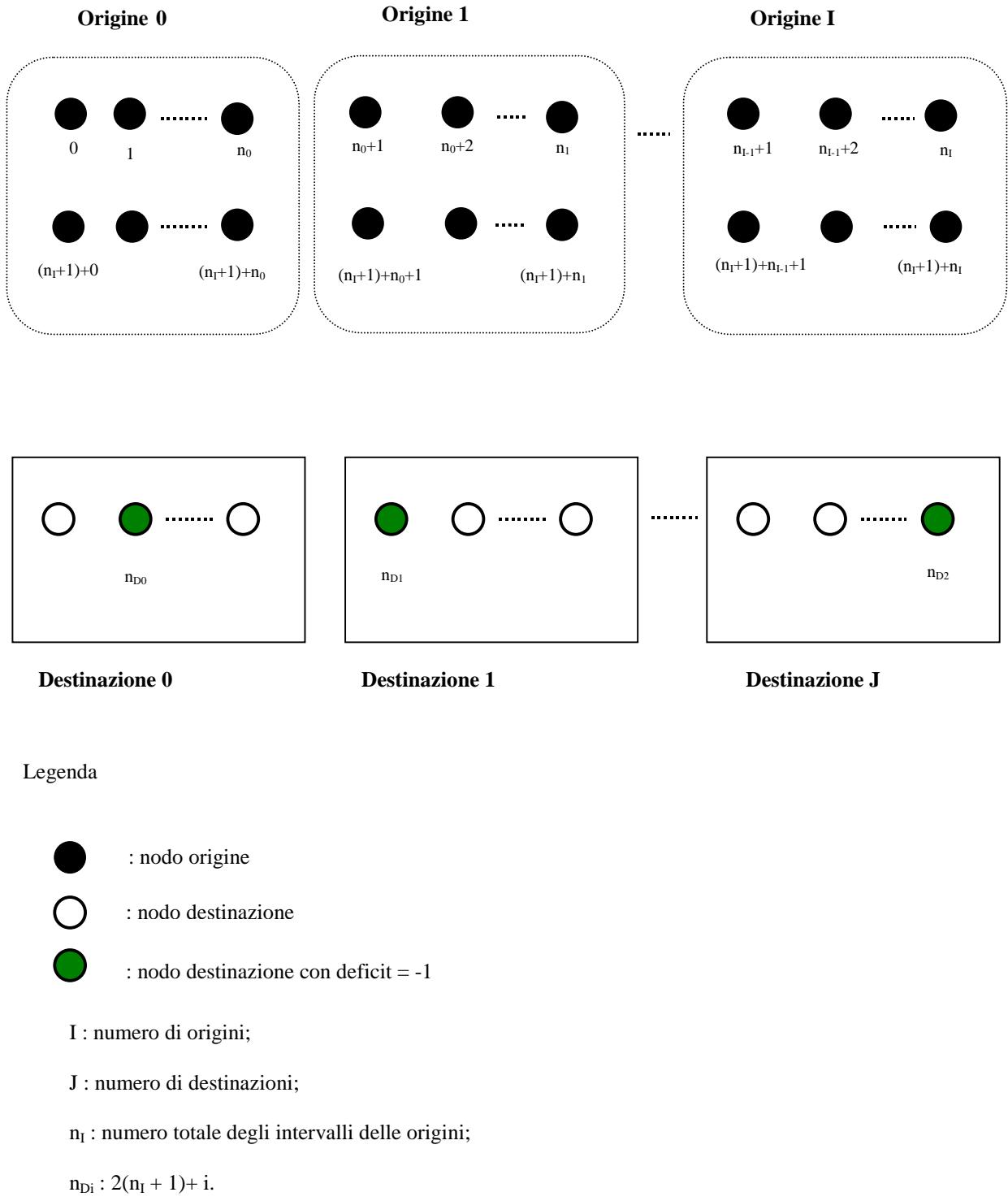
[Fig._: Situazione iniziale]

Dopo la suddivisione, delle giornate lavorative delle origini e delle destinazioni, in intervalli la situazione è quella in Fig._ , da notare che per le origini va considerato anche il “processo di caricamento”, pertanto vengono aggiunti i “nodi di carico”.



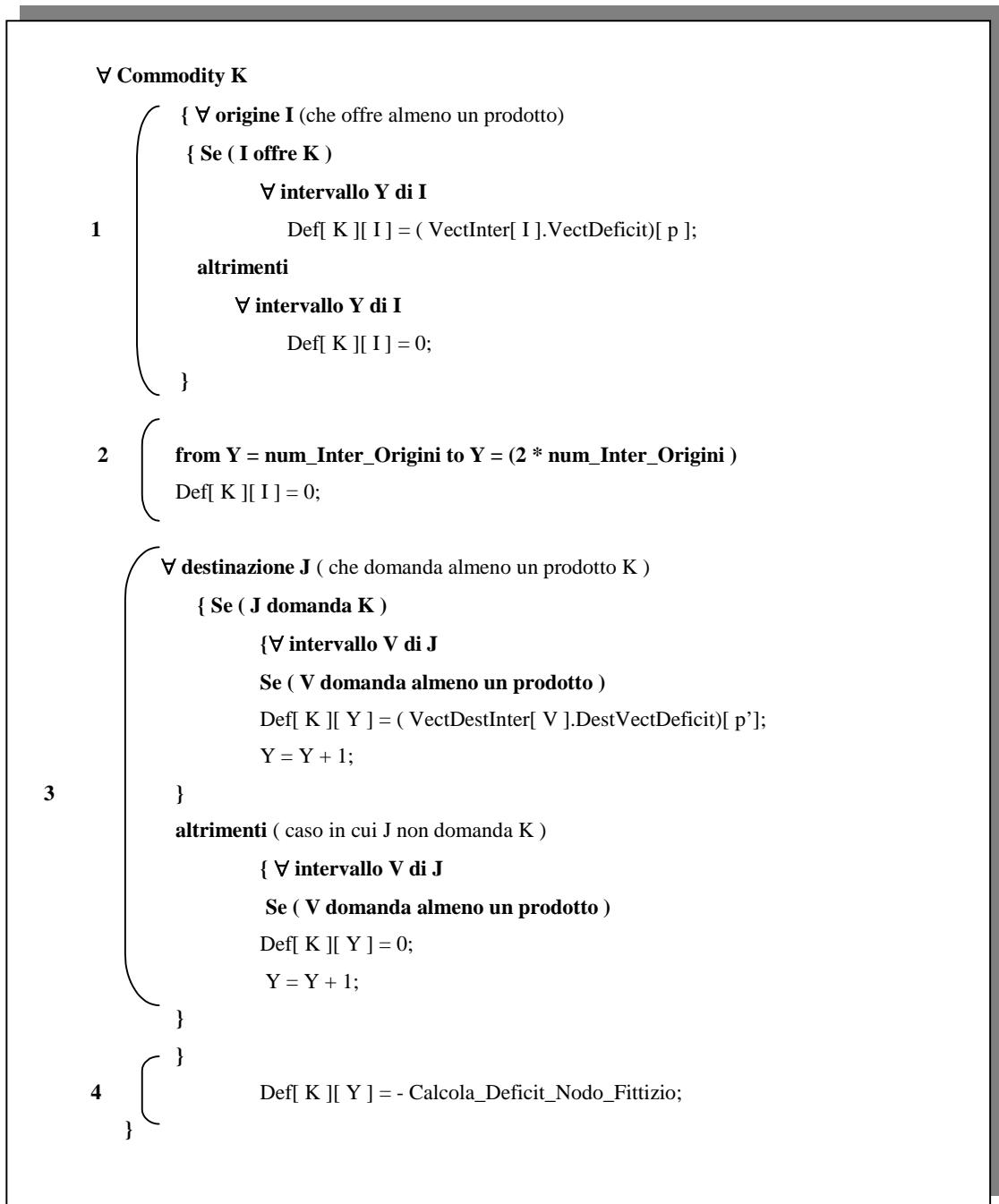
[Fig._ : Suddivisione in intervalli e aggiunta dei “nodi di carico”]

Il passo successivo è quello di attribuire una numerazione a questi nodi, ma poiché nel modello matematico è previsto il trasferimento delle unità di prodotto da un intervallo all’intervallo successivo di una stessa origine, gli intervalli delle origini andranno considerati tutti, mentre, a destinazione, considereremo solo gli intervalli in cui è stato stabilito dal procedimento di distribuzione della domanda, che dovrà arrivare almeno una unità di un qualsiasi prodotto richiesto; questo significa che un intervallo X di una destinazione J diventa un nodo del grafo se e solo se esiste almeno un “1” nel suo vettore dei deficit, come si può vedere dalla Fig._. Gli intervalli delle origini sono numerati consecutivamente nel vettore VectInter e tale numerazione continua ad essere la stessa per i nodi del grafo. Dall’ultimo intervallo dell’ultima origine inizia la numerazione dei “nodi di carico”. Per sapere a quale intervallo y si riferisce il nodo di carico y' è sufficiente il seguente calcolo: $y = y' \bmod (n_I + 1)$. Infine, a partire dall’ultimo “nodo di carico” inizia la numerazione dei nodi destinazione. L’ultimo nodo del grafo sarà il nodo fittizio a verranno collegati gli ultimi intervalli di ogni origine.



3.3.1 Creazione della matrice dei deficit

Per la creazione della matrice di deficit bisogna considerare i deficit di ogni intervallo, sia delle origine che delle destinazioni. La procedura è la seguente:



Con il **ciclo 1** vengono assegnati i deficit dei nodi relativi alle origini. L'indice p usato nell'assegnamento viene restituito dalla procedura che andrà a controllare se il prodotto concidente con la commodity K è offerto dall'origine I: più precisamente p indica il numero della commodity del prodotto all'interno della singola origine ed è in quella posizione del vettore dei deficit che troveremo l'offerta del singolo nodo. Dopo i deficit dei nodi origine, con il **ciclo 2** vengono assegnati i deficit ai “nodi di carico”: qui la variabile num_Inter_Origini è il numero totale degli intervalli delle origini. Con il **ciclo 3** vengono assegnati i deficit dei nodi destinazione. L'indice p'

come nel caso delle origini, verrà restituito dalla procedura che andrà a calcolare se il prodotto relativo alla commodity K è domandato dalla destinazione J. Infine, con l'**istruzione 4** viene calcolata la domanda del nodo fittizio come differenza tra tutto ciò che viene offerto e tutto ciò che viene domandato del prodotto relativo alla commodity K.

3.3.2 Inserimento degli archi

Una volta stabilita la numerazione dei nodi, andiamo ad inserire, nel grafo, gli archi con i rispettivi costi e capacità. Per ogni nodo che rappresenta un intervallo dell'origine vengono messi due archi: uno che collega il nodo al successivo, al quale viene assegnato costo zero, capacità individuale e capacità mutua infinito; l'altro che invece lo collega al nodo di carico, soggetto a capacità mutua uguale a 1. Inoltre, per l'ultimo nodo di ogni origine, viene messo un arco per il collegamento con il nodo fittizio. In totale gli archi che riguardano i nodi origine sono $m' = \text{Num_Inter_Origini} * 2$.

La procedura per l'inserimento di questi archi è la seguente:

```

J = 0;
 $\forall$  origine I ( che offre almeno un prodotto K )
{
     $\forall$  intervallo Y di I ( tranne l'ultimo )
    /* ARCO DI TRASFERIMENTO */
    E[ J ] = Y;
    S[ J ] = Y + 1;
    CapTot[ J ] = INFINITO;
    J = J + 1;
    /* ARCO DI CARICO */
    E[ J ] = Y;
    S[ J ] = num_Inter_Origini + Y ;
    CapTot[ J ] = 1;
    J = J - 1;
     $\forall$  commodity K
    { Se (origine I offre K )
        Cost[ K ][ J ] = 0; /* arco di trasferimento */
        Cap[ K ][ J ] = INFINITO;
        J = J + 1;
        Cost[ K ][ J ] = 0; /* arco di carico */
        Cap[ K ][ J ] = 1;
    altrimenti
        Cost[ K ][ J ] = INFINITO; /* arco di trasferimento */
        Cap[ K ][ J ] = 0;
        J = J + 1;
        Cost[ K ][ J ] = INFINITO; /* arco di carico */
        Cap[ K ][ J ] = 0;
    }
}

```

```

J = J - 1;
}

J = J + 2;
} /* fine del ciclo sugli intervalli */

/* ARCHI DI COLLEGAMENTO CON IL NODO FITTIZIO */

E[ J ] = Y;
S[ J ] = n - 1; /* dove n è il numero totale dei nodi */
CapTot[ J ] = INFINITO;
J = J + 1;

E[ J ] = Y;
S[ J ] = num_Inter_Origini + Y;
CapTot[ J ] = 1;
J = J - 1;

forall commodity K

{ Se (origine I offre K)

    Cost[ K ][ J ] = 0;
    Cap[ K ][ J ] = INFINITO;
    J = J + 1;
    Cost[ K ][ J ] = 0;
    Cap[ K ][ J ] = 1;

altrimenti

    Cost[ K ][ J ] = INFINITO;
    Cap[ K ][ J ] = 0;
    J = J + 1;
    Cost[ K ][ J ] = INFINITO;
    Cap[ K ][ J ] = 0;
    J = J - 1;
}

J = J + 2;
} /* fine del ciclo sulle origini */

```

Restano da inserire gli archi tra i nodi di carico e i nodi destinazione(archi origine-destinazione) che rappresentano tutti i viaggi che possono essere fatti. Contemporaneamente alla creazione di questi archi viene creato il VettoreViaggi che sarà il punto di partenza per la risoluzione del problema di assegnamento. Ogni elemento di questo vettore è un oggetto di tipo ViaggioCarico, che verrà descritto nel prossimo paragrafo, contenente le informazioni che contraddistinguono ogni viaggio come ad esempio l'origine della partenza, la destinazione di arrivo e il prodotto trasportato. La procedura per l'inserimento di questi archi è la seguente:

```

v = 0;
forall origine I ( che offre almeno un prodotto )
{
    forall intervallo Y di I
        num = -1;
        forall destinazione J
            Se ( ( J domanda almeno un prodotto ) and
                  ( I offre almeno un prodotto di quello che domanda J ) )
            {
                forall intervallo Y' di J
                    Attesa = Calcola_Tempo_Attesa_Viaggio( Y, Y' )
                    Se ( Non_Tutti_Zero_Deficit ( Y' )
                    {
                        num = num +1;
                        Se ( Attesa le Max_Attesa )
                            E[ J ] = num_Inter_OriginI + Y;
                            S[ J ] = ( 2 * num_Inter_OriginI ) + num;
                            CapTot[ J ] = INFINITO;
                        forall commodity K
                        {
                            Se ( ( Y offre K ) and ( Y' domanda K ) )
                                Cost[ K ][ J ] = CalcolaCosto(Attesa, I, J, K );
                                Cap[ K ][ J ] = 1;
                                /* CREAZIONE VETTORE VIAGGI */
                                VettoreViaggi[ v ].ORI = I;
                                VettoreViaggi[ v ].DES = J;
                                VettoreViaggi[ v ].InizioIntORI = OraInizioIntervallo( Y );
                                VettoreViaggi[ v ].FineIntDES = OraFineIntervallo( Y' );
                                VettoreViaggi[ v ].Comm = K;
                                VettoreViaggi[ v ].CamionUsato = camion che trasporta la comm. K

                                v = v + 1;
                            altrimenti
                                Cost[ K ][ J ] = INFINITO;
                                Cap[ K ][ J ] = 0;
                                J = J + 1;
                            }
                        }
                    /* fine ( Non_Tutti_Zero... )
                } /* fine ( Se ( ( J domanda almeno... ) */
            }
        }
    }
}

```

altrimenti

```
{  
  Se ( J domanda almeno un prodotto )  
    forall intervallo Y' di J  
      Se ( Non_Tutti_Zero_Deficit ( Y' ) )  
        num = num +1; /* è necessario per numerare correttamente i nodi destinazioni*/  
    }  
  } /* fine (forall origine I ...) */
```