UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"
FACOLTÀ DI INGEGNERIA
Dipartimento di Progettazione Aeronautica

**Dottorato di Ricerca in Ingegneria Aerospaziale,**
**Navale e della Qualità**
**Indirizzo Gestione della Qualità Totale**

# Quick response nella pianificazione della produzione: applicazione del bootstrap nella simulazione event-driven

Coordinatore
Prof. Ing. Antonio Moccia

Tutor
Prof. Ing. L.C. Santillo
Prof. Ing. M. Staiano

Dottorando
Ing. Guido Guizzi

XVIII Ciclo di Dottorato

## Ringraziamenti

Desidero ringraziare la prof. Tina Santillo ed il prof. Vincenzo Zoppoli per avermi guidato e per la fiducia e la stima dimostratami in questi anni di dottorato.
Ringrazio, inoltre, il prof. Antonio Lanzotti ed il prof. Michele Staiano, per il supporto ed il continuo stimolo ad approfondire tematiche d'interesse per l'attività di ricerca, ed il prof. Pasquale Erto, per le puntuali ed acute osservazioni in merito all'attività svolta.
Un profondo ringraziamento va al prof. Roberto Revetria per l'aiuto fornitomi e l'amicizia dimostratami.
Un grazie agli amici, ancor prima che colleghi, Mosè Gallo, Teresa Murino, Pasquale Buccione, Luigi Guerra, Annarita Franco, Pino Converso, Elpidio Romano, Pasquale Zoppoli e Paolo D'Ambrosio per i piacevoli scambi di idee sulle tematiche oggetto di studio.

Il grazie più grande va al mio papà Guglielmo e a mia moglie Stefania, per aver sempre creduto in me e non avermi mai fatto mancare il loro sostegno e il loro amore.

A mia madre e ad Anna Maria

*Chi è maestro nell'arte*
*di vivere fa poca*
*distinzione tra il proprio*
*lavoro e il proprio gioco,*
*la propria fatica e il*
*proprio divertimento, la*
*propria mente e il*
*proprio corpo, il proprio*
*studio e il proprio svago,*
*il proprio amore e la*
*propria religione.*
*Quasi non sa quale sia dei due.*
*Persegue semplicemente*
*il proprio ideale di*
*eccellenza in tutto quello*
*che fa, lasciando agli*
*altri decidere se stia*
*lavorando o giocando.*
*Ai suoi occhi lui sta*
*sempre facendo*
*entrambi.*

Budda

# Indice

# Premessa

Oggi i mercati stanno duramente mettendo alla prova la capacità di tenuta di una intera classe imprenditoriale. Il "modello" di impresa reattiva, vitale e, quindi, competitiva è certamente costituito da tanti tasselli: ricerca & innovazione, informatica, credito, acquisti, qualità, misure e controlli, formazione, ed altro. Per la aziende manifatturiere certamente la competitività nel medio lungo termine è legata alla loro capacità di innovare processi/prodotti e nello sviluppare opportune azioni di marketing. Nel breve medio termine, certamente, la competività può passare in un recupero di efficienza dei processi interni e di riorganizzazione. In tal senso si ritiene che miglioramenti nella programmazione della produzione possano consentire sia un recupero di efficienza che un miglioramento in termini di soddisfazione dei clienti.

La pianificazione della produzione è una delle fasi più impegnative e complicate della gestione di un' impresa. A suo supporto vi sono una serie di tecniche e metodologie di risoluzione che, malgrado risultino piuttosto consolidate e affidabili, presentano diverse ipotesi non sempre verificate nelle realtà aziendali. La complessità delle interazioni tra gli elementi della catena del valore rende i risultati derivanti dalle tecniche di pianificazione statica, attualmente in uso, poco affidabili. Infatti, vi è un aumento esponenziale della difficoltà computazionale legato ai diversi aspetti che man mano occorre tenere in considerazione, tra i quali: tipologia di sistema produttivo (job shop, flow shop), numero di macchine, utilizzo delle risorse, cicli tecnologici, schedulazione dei job, varietà dei prodotti (sia in termini qualitativi che in termini quantitativi), rispetto dei tempi di produzione, vincoli di *due date* ed altro.

La necessità di legare queste caratteristiche diverse e spesso in contrasto tra loro, è uno dei fattori più critici da realizzare ma, allo stesso tempo, di fondamentale importanza. Infatti, è estremamente complesso riuscire a trovare un algoritmo capace di mediare e dare una soluzione ottimale e completa a tale problema.

L'obiettivo principale della Tesi di Dottorato è stato quello di proporre un approccio alla pianificazione della produzione mediante simulazione.

Nell'ambito del paradigma lean, è stato sviluppato un approccio alla pianificazione della produzione per sistemi job-shop che finalizzato alla massimizzazione del livello di servizio al cliente senza aumentare il livello dei magazzini, ma organizzando opportunamente la produzione in ottica just in time. La complessità delle interazioni tra gli elementi della catena del valore rende i risultati derivanti dalle tecniche di pianificazione attuali poco affidabili. L'utilizzo sinergico di modelli di lotsizing, modelli di simulazione del processo produttivo e tecniche statistiche di progettazione degli esperimenti (DOE), validazione ed analisi della varianza (ANOVA) consente di superare i limiti delle tecniche attualmente in uso e permette di introdurre nuove variabili nel processo di pianificazione. Infatti è stata introdotta la possibilità di considerare, in fase di pianificazione, componenti provenienti da prodotti a fine vita (EOL).

I principali risultati ottenuti hanno riguardato i modelli di pianificazione, la sinergia tra tali modelli ed i simulatori trace-driven, i metodi per la validazione dei modelli di simulazione. I risultati sono stati originali in quanto hanno permesso, per quanto concerne i modelli di pianificazione, di progettare un modello di pianificazione operativa della produzione che dia la possibilità di utilizzare sia di materia prima nuova che di materia prima secondaria proveniente da processi di reverse logistic; per quanto concerne i metodi di validazione dei modelli di simulazione, di valutare la possibilità di utilizzare tecniche alternative a quelle tradizionali che già per sistemi diversi da quello oggetto di studio hanno dato risultati soddisfacenti. Possono inoltre avere significative ricadute in campo industriale e applicativo grazie al superamento dei limiti delle tecniche di pianificazione tradizionale; infatti i sistemi ERP realizzati dai principali produttori (SAP, BAAN, ORACLE) utilizzano per la pianificazione modelli statici in cui vengono definiti i vincoli del problema mediante la codifica di opportune regole. Purtroppo tali sistemi tendono a sovrastimare l'effettiva capacità produttiva dell'azienda. Infatti tale capacità può variare significativamente in funzione del mix produttivo pianificato e in funzione dei volumi. Tale errore risulta limitato se il sistema produttivo è di tipo flow-shop, viceversa per i sistemi job-shop è un problema estremamente sentito. Vista l'impostazione metodologica utilizzata, sarebbe auspicabile interfacciare il modello di pianificazione proposto con il database transazionale

aziendale permettendo di assicurare “due date” poco variabili e definibili già all’emissione dell’ordine di vendita.

# Introduzione

Un processo produttivo, in quanto sistema complesso, è governato da molteplici variabili che hanno un certo grado di aleatorietà. Nel caso più generale tale aleatorietà può essere funzione anche del tempo e dello stato in cui si è trovato il sistema in istanti precedenti. In tal senso è possibile pensare di schematizzare ciascuna variabile come una funzione aleatoria di una variabile deterministica t.
Questo ulteriore approccio apre un insieme di scenari in cui la schedulazione non ha più come obiettivo solo quello di realizzare al minimo costo i prodotti al tempo giusto, ma di fornire una soluzione robusta che risulti poco sensibili alle fluttuazioni delle variabili del processo.
L'obiettivo del lavoro è quello di proporre, nell'ambito del paradigma lean, un approccio alla pianificazione della produzione per sistemi job-shop che massimizzi il livello di servizio al cliente senza aumentare il livello dei magazzini, ma organizzando opportunamente la produzione in ottica just in time.
In tal modo, a partire da una possibile soluzione di scheduling, se ne verifica la fattibilità grazie al simulatore con conseguente determinazione della significatività statistica degli output della simulazione ed infine si esegue un'analisi dei dati per verificare l'eventuale influenza di variabili gestionali sugli indici di performance (Fig. I).
Per raggiungere tale obiettivo è necessario, in una prima fase, determinare il momento in cui far partire gli ordini di produzione su base giornaliera rispettando i vincoli di capacità delle risorse (manodopera e macchine).
Per individuare i lotti di produzione giornalieri, dopo un'approfondita analisi bibliografica, è stato scelto un algoritmo proposto da Sürie di CLSPL ( Capacitated Lot-sizing Problem with Linked Lot-Sizes). Tale algoritmo, ha come funzione obiettivo, la minimizzazione dei costi di produzione, intesi come costi di mantenimento a magazzino, costi di setup e costi di straordinario previsti sulle diverse risorse.

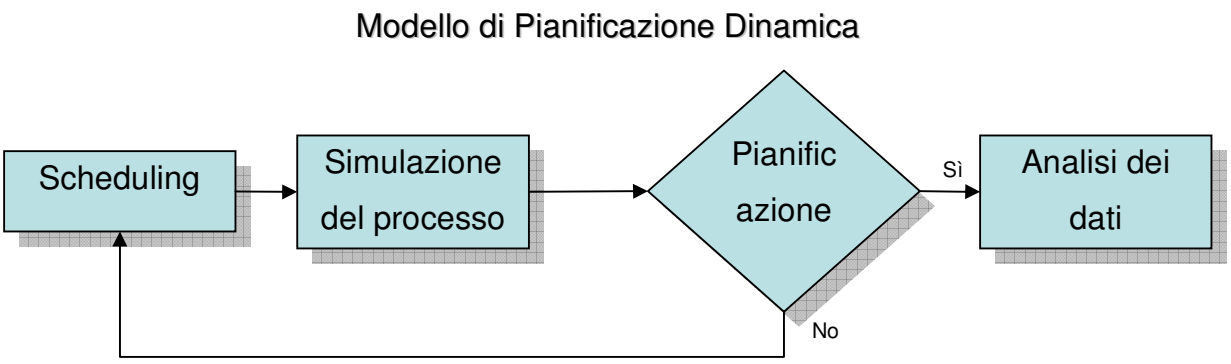



**Fig. I. Modello di Pianificazione Dinamica**

L'orizzonte di pianificazione T viene diviso in intervalli *t*, ed i lotti di produzione vengono dimensionati e schedulati in modo tale che i prodotti siano disponibili per un dato periodo *t*, soddisfacendo in tal modo il vincolo di *due date* e rispettando i tempi di produzione.

Il modello CLSPL è stato migliorato, introducendo delle ulteriori variabili nella funzione obiettivo per quanto concerne la possibilità di backlog, ovvero ordini che vengono consegnati in ritardo, e ricorso a terzisti per esternalizzare parte della produzione. In tal senso è stato associato un costo al backlog inteso come penale da dover pagare per la mancata consegna ed un costo per la produzione delegata a terzi. Per quest'ultima sono previsti anche dei vincoli che consentono di impostare la percentuale della produzione del prodotto *j* che è possibile esternalizzare.

Tale modello è stato implementato mediante il software LINGO della Lindo System e tutti dati necessari per il corretto funzionamento vengono forniti mediante apposito database relazione sviluppato in ambiente MS Access.

La seconda fase del lavoro è stata quella di sviluppare un modello di simulazione per verificare la soluzione proposta dal modello di scheduling precedente. La simulazione è una delle tecniche più importanti a servizio delle imprese. Tramite essa le aziende possono risparmiare ingenti quantitativi di denaro e di tempo simulando gli effetti derivanti da possibili scelte, senza che vi siano ripercussioni sul sistema produttivo. Valutando i risultati della simulazione può essere presa una decisione che si basi su dati più concreti di semplici supposizioni soggettive.

In particolare, al fine di generalizzare il modello di simulazione e quindi renderlo applicabile ai diversi contesti industriali, si è deciso di sviluppare un meta-modello di simulazione che schematizzi i principali stati di un ordine di produzione e generi automaticamente un modello di simulazione del particolare processo produttivo attingendo le necessarie informazioni da una base dati appositamente progettata.

La base dati per la simulazione prevede l'utilizzo di diverse tabelle contenenti molteplici set di dati raggruppabili nelle seguenti categorie:

- Dati relativi alle risorse;
- Dati relativi ai job;
- Dati relativi alle operazioni.

Il primo set di dati fornisce informazioni sulle macchine disponibili, le risorse umane ad esse assegnate, le relative capacità, i tempi di setup, la presenza di eventuali operazioni di controllo di qualità e i costi, compresi anche i costi di straordinario. Questo tipo di informazioni risultano essere statiche, poiché relative alla struttura dell'impianto di produzione.

I dati relativi ai job, invece, forniscono informazione sui job attuali e, in tal senso, subiscono continui aggiornamenti. In particolare, per ogni prodotto è definita la distinta base, la specifica degli ordini di acquisto e di produzione, il ciclo tecnologico, i lead time, le *due date*.

Infine, per quanto concerne i dati relativi alle operazioni, per il magazzino centrale e i magazzini logici definiti per ciascun ordine di produzione, sono indicate tutte le operazioni di carico e scarico con i relativi dati in termini di quantità movimentate di ciascun componente, ordini ai quali tali componenti sono destinati e date di disponibilità, in modo da poter gestire correttamente gli impegni di prodotti e componenti.

Oltre ai vantaggi ampiamente noti derivanti dall'impiego di database relazionali per la gestione dei dati, tra i quali la minimizzazione degli errori di inserimento dati e il facile aggiornamento, essendo il modello di simulazione sviluppato in maniera automatica in base alle informazioni presenti sul database, è possibile apportare modifiche ed aggiornare il modello del processo produttivo facilmente anche nel caso di problemi di grandi dimensioni (Fig. II).

Per quanto concerne la simulazione, l'obiettivo è quello di modellare e simulare una produzione di tipo job shop in maniera efficiente e flessibile. Inoltre, grazie alla definizione del modello del processo produttivo mediante database, è possibile simulare diversi scenari alternativi scegliendo la configurazione che risulta più flessibile ed adeguata a rispondere al trade off tra costi e livello di servizio offerto.

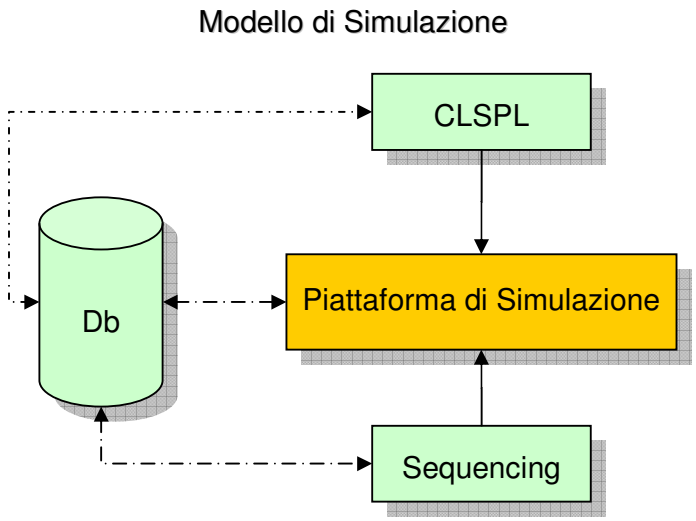



**Fig. II *Modello di Simulazione***

Il meta-modello (piattaforma) di simulazione è stato realizzato utilizzando il software ARENA della Rockwell. Tale software, essendo un simulatore general purpose, permette di integrare la semplicità di utilizzo, tipica dei simulatori di alto livello, con la flessibilità necessaria a soddisfare specifiche esigenze, quali l'implementazione di algoritmi decisionali complessi o l'acquisizione di dati da applicazioni esterne, ottenuta impiegando linguaggi di programmazione come il C++ e il Visual Basic. Infine, tale software fornisce numerosi strumenti di rappresentazione grafica e la possibilità di sviluppare specifici reports al fine di facilitare l'analisi e la rappresentazione dei risultati della simulazione. Tali informazioni sono direttamente riportate sul database. In particolare, sono specificati i tempi di inizio e fine di ciascuna simulazione, la data di disponibilità

di ciascun ordine, le informazioni relative al tempo di attraversamento degli ordini sulle singole macchine, la produttività e gli eventuali scarti.

Il modello di simulazione è stato sviluppato al fine di ottenere un sistema di pianificazione della produzione che non sia focalizzato esclusivamente sulla saturazione della capacità produttiva e la minimizzazione dei costi, ma che consideri la rilevanza della soddisfazione del cliente, permettendo di assicurare *"due date" poco variabili e definibili già all'emissione dell'ordine.* Al sistema è assegnato in input un piano della produzione, a partire dal portafoglio ordini, secondo l'algoritmo CLSPL. A questo punto, impiegando le informazioni fornite, il modello di simulazione stesso effettua il sequencing degli ordini sulla base di algoritmi di ottimizzazione che minimizzano il makespan.

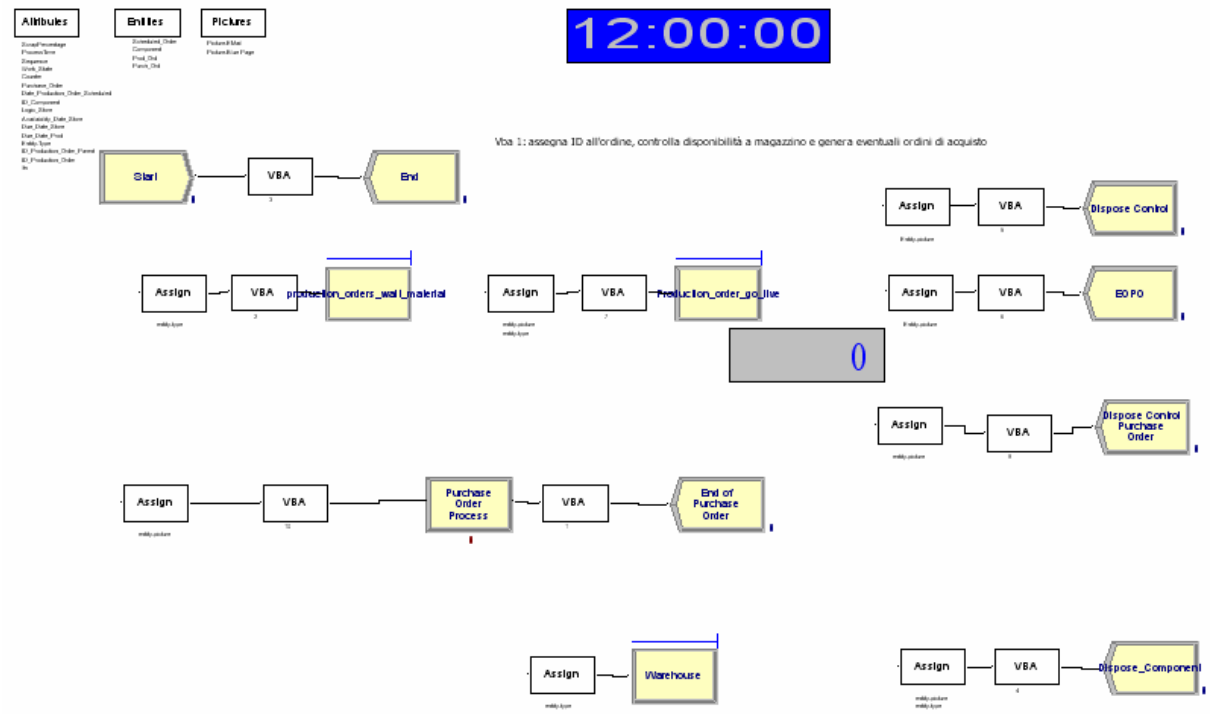



**Fig. III Meta-modello di simulazione sviluppato in ambiente ARENA**

Il sequenziamento ottimale viene stabilito durante la simulazione mediante un apposito algoritmo di sequencing. La ricerca bibliografica svolta ha evidenziato la presenza in letteratura di numerosi modelli di sequencing nel caso di sistemi flow-shop. Per quanto concerne i sistemi job-shop sono stati trovati algoritmi

euristici che forniscono soluzioni ottimali. In particolare è stato scelto un algoritmo che genera soluzione con particolari proprietà:

- Soluzioni *attive*: per anticipare il completamento di una qualsiasi operazione, è necessario alterare la successione di operazioni su qualche macchina provocando così, un ritardo nel completamento di altre operazioni;
- Soluzioni *senza ritardo o nondelay:* per le quali non si verifica mai che una macchina pur potendo effettuare una operazione resti inattiva.

In generale si può dimostrare che una soluzione senza ritardo è certamente attiva e che la soluzione ottima di un problema job shop con obiettivi regolari (ad esempio *minimizzazione del makespan*) sarà sicuramente una schedulazione attiva. Questo metodo consta nell'individuare, utilizzando le matrici dei tempi di processamento e di routing, l'insieme delle operazioni ammissibili. Nota la sequenza tecnologica, il modello proposto è un metodo iterativo che partendo dalla prima operazione di ogni job, assegnandolo ad una macchina e seguendo le informazioni delle matrici di routing e di processamento, assegna, ad ogni passo, la lavorazione di uno dei job dell'insieme alla macchina, facendolo passare alla operazione successiva nell'insieme delle operazioni ammissibili della seconda iterazione e così via, fino a quando non saranno stati assegnati tutti i job per tutte le operazioni, alle macchine. L'algoritmo proposto è stato implementato nel linguaggio di programmazione Visual Basic.

L'impostazione utilizzata per la piattaforma di simulazione consente, comunque, in qualsiasi momento di sostituire specifici moduli senza alterare il modello complessivo di pianificazione dinamica proposto; pertanto eventuali miglioramenti dei diversi moduli utilizzati possono essere facilmente implementati sulla piattaforma.

Al termine della simulazione, è necessario effettuare un'analisi dei dati ottenuti. In questa fase si è ricorso, attestata la significatività dei dati statistici, al metodo Bootstrap al fine di ridurre il numero di simulazioni necessarie per un'analisi adeguata. Il metodo Bootstrap, in particolare, è una tecnica di ricampionamento che permette, attraverso il calcolo dell'errore standard connesso alle statistiche di interesse e la realizzazione di test di significatività, di quantificare l'incertezza

connessa ai risultati sperimentali ottenuti dalla simulazione. Il vantaggio di tale metodo è quello di poter effettuare tali valutazioni con un campione di dimensioni ridotte e senza dover assumere come distribuzione di partenza della popolazione una Normale. In particolare, tale ipotesi risulta piuttosto restrittiva soprattutto nel caso di simulazione di sistemi reali. Inoltre, poiché il metodo Bootstrap assume quale popolazione di partenza il campione originario, è possibile effettuare un'accurata analisi statistica dei risultati della simulazione impiegando un numero piuttosto contenuto di iterazioni, riducendo in tal modo in maniera significativa il tempo di simulazione.

In particolare, tale metodo è stato impiegato nell'ambito della simulazione, al fine di effettuare la validazione di modelli di simulazioni in tempi più rapidi rispetto alle tecniche tradizionali e prescindendo dall'assunzione di normalità delle distribuzioni delle variabili. In particolare, Kleijnen, Cheng e Bettonvil (2000) hanno dimostrato la validità dell'impiego di tale metodo nel caso di simulazioni trace-driven. Infatti, le tecniche tradizionali confrontano il comportamento del sistema reale e di quello simulato, entrambi soggetti a input comuni, sulla base dei rispettivi indici di performance X e Y, nell'ipotesi, non solo che questi risultino i.i.d., ma anche che le coppie (Xi,Yi) per ogni subrun i-esimo siano normali bivariate. L'impiego del metodo Bootstrap pur considerando le coppie (Xi,Yi) correlate, permette di non assumere l'ipotesi di normalità delle distribuzioni, ovvero di effettuare la validazione del modello con un numero di subrun molto ridotto. In particolare, tale tecnica può essere impiegata in sostituzione ai tradizionali metodi di valutazione basati sui test t-student o su quelli distribution-free nel caso di un numero di replicazioni molto ridotto. La validazione del modello è effettuata sulla base di sei statistiche provenienti dalla pratica e dalle teorie relative all'analisi statistica, differenziando le procedure Bootstrap in funzione del numero di replicazioni effettuate, fissato il numero n di subrun per ciascuna replicazione.

Per quanto concerne la determinazione del numero di campioni Bootstrap necessari, numerosi studi sono stati effettuati in merito. In particolare, Kleijnen, Cheng e Bettonvil hanno definito come numero minimo di b il valore $(2/\alpha)-1$ per

una sola replicazione e (2s/ α)-1 nel caso di s replicazioni sulla base della disuguaglianza di Bonferroni.

Infine, il metodo Bootstrap permette di effettuare la valutazione degli intervalli di confidenza utilizzando la metodologia tradizionale se la distribuzione del campione di partenza è prossima alla Normale e il bias è ridotto. In tal caso l'intervallo di confidenza per la media campionaria risulta pari a: $\bar{x} \pm t^*_{\alpha/2,n-1} \cdot SE$

dove *SE* è la deviazione standard della distribuzione bootstrap.

Se i campioni sono di ridotte dimensioni e le ipotesi di base non sono rispettate è possibile ricorrere al metodo bootstrap bias-corrected accelerated (BCa). Gli estremi degli intervalli di confidenza BCa sono percentili della distribuzione bootstrap modificati per correggere il bias e la skewness della distribuzione. Ad esempio, se la statistica è molto ampia o la distribuzione è skewed verso destra, il limite dell'intervallo di confidenza è traslato verso sinistra. Dopo l'applicazione del metodo BCa, il calcolo dell'intervallo di confidenza viene effettuato allo stesso modo del caso precedente.

Nella parte conclusiva del lavoro viene presentata un'applicazione completa della simulazione ad un modello di impianto per la produzione di congelatori. In tale applicazione, grazie all'utilizzo di tecniche DOE ed ANOVA, viene evidenziata l'influenza di alcuni parametri del processo produttivo sulla produttività del sistema. I risultati ottenuti sono piuttosto intereressanti e fanno percepire le enormi potenzialità dell'approccio proposto.

# 1.Lean thinking

## 1.1. Introduzione

Nelle vicende degli ultimi anni è facile cogliere un'esigenza sempre più diffusa nel mondo industriale: il modello tradizionale d'impresa non risulta più adatto al contesto in cui si trovano le aziende, sempre più sottoposte ad una pressione competitiva spietata e di diversa natura. Si è avvertita, quindi, la necessità di ripensare il modello d'impresa.

Questa esigenza costituisce il motivo dominante di tutta la letteratura di management e dei mezzi di comunicazione di questi anni. L'aumento della pressione competitiva è certamente difficile da quantificare e misurare, ma può essere pensata giustificabile da almeno quattro fattori [29]:

- l'*abbattimento delle barriere geografiche*, che ha ampliato in modo notevole il numero dei concorrenti con cui ogni azienda è costretta a misurarsi;
- le *liberalizzazioni*, che dappertutto hanno investito alcuni settori un tempo a gestione monopolistica (telecomunicazioni, energia elettrica, ecc.);
- una rapidissima *innovazione*, che ha l'effetto di mettere in campo sempre nuovi prodotti e nuovi concorrenti con cui misurarsi;
- infine, l'avvento e la *diffusione di internet* che, oltre ad ampliare gli orizzonti geografici raggiungibili dalle imprese e quindi il numero di potenziali concorrenti, accresce notevolmente la possibilità di ricerca e confronto dei consumatori aumentando il loro potere contrattuale.

La pressione competitiva, però, non è soltanto più intensa ma è anche molto diversa. Ciò può essere attribuito a diversi fattori, il più importante dei quali è sicuramente l'*innovazione continua*; attraverso la riduzione dei cicli di vita dei prodotti, infatti, si riduce il tempo utile di sfruttamento, producendo un'accelerazione dei tempi d'azione. La riduzione dei margini di profitto innesca, poi, varie reazioni come la ricerca di maggiore efficienza e di riduzione dei costi, di un contatto più profondo con i clienti attraverso prodotti e servizi differenziati per dare "valore aggiunto", ecc.

Poi ci sono i comportamenti dei potenziali consumatori che, attraverso le tecnologie internet, diventano soggetti in continuo movimento, sempre meno fedeli e pronti a cambiare fornitore.
Un altro fattore è senza dubbio anche la grande mobilità acquistata in alcuni settori dal mercato del lavoro che obbliga le imprese ad uno sforzo continuo di ricerca del personale, di formazione e di stabilizzazione. La grande rapidità acquistata dalle relazioni sociali ed economiche, infatti, fa diventare fattori critici di successo la velocità di acquisizione delle informazioni, la rapidità di reazione ai cambiamenti e la tempestività di risposta.
Alla luce di queste riflessioni, sorge la necessità di trovare nuovi modelli d'impresa che siano in grado di far fronte a questa situazione; capaci, cioè, di essere creativi e innovativi, in grado di percepire e gestire i cambiamenti. Ma l'individuazione di un nuovo modello adatto alle nuove condizioni competitive è un compito molto difficile; si tratta di un processo complesso fatto di intuizioni imprenditoriali, di sperimentazioni continue dentro l'impresa, di riflessioni teoriche di studiosi in materia, di imitazioni e di errori. E non c'è probabilmente una risposta migliore in senso assoluto. Alcuni autori per definire tale modello individuano diversi filoni di lavoro come:

1. l'aumento dell'ampiezza del controllo e la riduzione dei livelli gerarchici; tale intervento non è risolutivo per quello che riguarda la creatività, la rapidità di risposta, il grado di coinvolgimento del personale, anche se gli effetti sono di solito positivi;
2. la riduzione della dimensione d'impresa; il ridimensionamento organizzativo può fare molto per conferire all'impresa la concentrazione e la snellezza richieste dal nuovo contesto competitivo;
3. la gestione con valori invece che la gestione con le regole; le imprese tradizionali adottano regole, istituzioni, manuali e procedure che spesso costituiscono vere e proprie barriere fra di loro ostacolando in maniera notevole il corretto funzionamento dell'impresa. È consigliabile abbattere le rigide divisioni di responsabilità e fare condividere al personale dell'azienda la stessa visione e gli stessi valori per permettergli di rispondere in modo autonomo e correttamente alle mutevoli condizioni competitive.

Tutti questi filoni sono parte integrante del modello universalmente noto come **Lean Thinking**, occidentalizzazione del leggendario "Toyota Production System" operata da due studiosi occidentali: il cosiddetto "*pensare snello"* non esprime concetti assolutamente nuovi, piuttosto può essere considerato come un'evoluzione dei modelli organizzativi che l'hanno proceduto (qualità totale, reingegnerizzazione dei processi, progettazione simultanea, gestione con i valori, ecc.) a cui riesce a dare una convincente sistematizzazione ed integrazione. Il *pensiero snello* è, quindi, prima di tutto una visione, poi è un modello tecnico-organizzativo e gestionale capace di ottenere elevate performance su più fronti; è una leva fondamentale per cambiare le regole della competizione e per acquisire rilevanti vantaggi competitivi; è un sistema che riesce ad utilizzare le risorse nel modo più conveniente e ad ottenere economie di scala attraverso stretti legami fra molte imprese a monte e a valle.
J.P. Womack e D.T. Jones, autori di "*The Machine that Changed the World*" e "*Lean Thinking*", hanno avuto il merito di estrarre da un approccio operativo e fortemente contestualizzato all'interno di uno specifico settore, quello automobilistico, un sistema organizzativo-gestionale, riassumendolo in una serie di principi applicabili in ogni settore dove vi sia la necessità di organizzare e gestire i processi.

## 1.2. Lean Manufacturing

La metodologia del Lean Thinking si sta rapidamente imponendo come uno degli strumenti più moderni ed efficaci per garantire alle aziende la flessibilità e la competitività che il moderno mercato richiede. Tale metodologia può essere applicata a tutte le aree aziendali.
Facendo riferimento all'area della produzione, assume sempre più importanza la cosiddetta **Produzione Snella** o **Lean Manufacturing**, nota anche come *TPS* (*Toyota Production System*), essendo legata ai sistemi produttivi nati in Toyota.
Il termine "Lean Manufacturing" descrive una filosofia gestionale che incorpora una serie di strumenti e tecniche da utilizzarsi nei processi aziendali per

ottimizzare il tempo, le risorse umane, le attività e la produttività, migliorando, nello stesso tempo, il livello qualitativo dei prodotti e servizi offerti al cliente [44].
Produrre in modo snello può essere definito come un sistema di riduzione degli sprechi in tutta l'organizzazione, dalla produzione fino agli uffici.
Il Lean Manufacturing, in quanto sistema integrato con altri (qualità, economico finanziario, ecc.), parte dall'alto con obiettivi tipici di business plan, per tramutarsi in progetti specifici di miglioramento, con focus in particolare nella produzione e nell'erogazione del prodotto/servizio.
Di fatto, applicare un sistema Lean Manufacturing non si discosta molto dall'applicare un sistema di gestione per la qualità: obiettivi a livello business vengono tramutati in azioni specifiche per i processi al fine di ridurre gli sprechi.
Il Lean Manufacturing, in ogni caso, deve comprendere e governare l'intero sistema di realizzazione del prodotto/servizio, gestendo i processi relativi al cliente, la progettazione e lo sviluppo, la produzione e tutta la catena di approvvigionamento (Supply Chain Management).

### *1.2.1. Strumenti e metodi di base della Lean Manufacturing*

Molti degli strumenti e dei metodi alla base di un sistema Lean Manufacturing sono stati ereditati dalle esperienze degli anni '80 effettuate dalle eccellenti aziende giapponesi, in particolar modo dalla "Toyota Motor Company". Gli strumenti e i metodi da seguire e utilizzare per raggiungere l'obiettivo di riduzione continua degli sprechi includono:

1. JIT: Just In Time;
2. Sistema "PULL" / Kanban;
3. Poka-Yoke;
4. SMED: Single Minute Exchange of Die;
5. TPM: Total Productive Maintenance;
6. Takt Time;
7. Heijunka;
8. Cellular Manufacturing;

9. Kaizen.

In particolare verranno analizzati solo i primi due strumenti in quanto impattano direttamente sulla organizzazione della produzione.

Il Just In Time[1], di seguito riportato con l'acronimo JIT, è una filosofia gestionale e una metodologia di gestione della produzione volta all'eliminazione di tutti gli sprechi di materiale, forza lavoro, spazio e tempo che si possono riscontrare all'interno di un sistema di realizzazione di un prodotto/servizio. Introdotto inizialmente dalla Toyota negli anni '60 e successivamente applicato con successo da molte altre imprese, per rispondere alle esigenze di un mercato saturo, instabile e dinamico, nonché caratterizzato da innovazioni tecnologiche, il JIT non si limita ad essere una tecnica di gestione, ma, come già detto, è una filosofia che richiede un cambiamento radicale nel modo di agire, di pensare e di comportarsi di tutti coloro che direttamente o indirettamente partecipano al processo produttivo.

Il principio logistico posto alla base della gestione dei flussi di materiali con la tecnica del JIT è quello secondo il quale *bisogna realizzare e consegnare, nella quantità e nella qualità necessaria e al minimo costo possibile, i prodotti finiti "appena in tempo" per essere consegnati ai clienti esterni dell'azienda*; tale concetto può essere anche internalizzato, nel senso che *ogni singola fase del sistema produttivo deve iniziare la lavorazione del prodotto nel momento in cui ne necessita la fase successiva.*

Ciò implica che:

1. I materiali grezzi devono giungere appena in tempo per essere lavorati;
2. I prodotti finiti da interno devono uscire dalle rispettive linee di lavorazione al momento opportuno per essere montate nei sottogruppi;
3. I sottogruppi devono giungere all'assemblaggio finale nel momento in cui devono essere utilizzati;
4. I componenti finiti da esterno devono entrare in azienda al momento opportuno per essere montati sul prodotto finale.

[1] Letteralmente "Appena in tempo", ma molto spesso tradotto meglio con il termine "Solo quando necessario".

Il modello JIT si basa su 6 aspetti fondamentali [167]:

5. **Just In Time Production**: produrre esattamente solo i quantitativi richiesti nel breve periodo e non anche quelli che, secondo le previsioni, si pensa di poter vendere in futuro. Infatti, si dice che la produzione è programmata *just in time* quando la consistenza dei magazzini di acquisto, di trasformazione e di vendita è ridotta ad un giorno;
6. **Stockless Production**: evitare l'accumulo di scorte, utilizzate non tanto per ragioni economiche (come economie di scala, risparmi sui costi di trasporto, ecc.), ma per coprire le inefficienze interne ed esterne;
7. **Eliminazione degli sprechi**, ovvero di tutte quelle attività e risorse che non si trasformano in valore aggiunto, come eccesso di produzione, bassa qualità, potenzialità, capacità e abilità non sviluppate, energie che si annullano a vicenda, ecc.;
8. **Produzione a flusso**: bisogna tendere verso la produzione a flusso, tipica delle industrie di processo, nella quale si passa dalle materie prime al prodotto finito senza interruzioni, evitando, così, i trasporti inutili e le *polmonature* intermedie;
9. **Pull System** (*sistema a trazione*): il materiale non avanza nel processo produttivo in base ad un programma di produzione stabilito sulla previsione della domanda, ma è richiamato ("*tirato*") direttamente dal reparto a valle che lo utilizza. Il ritmo di ogni reparto, quindi, viene imposto dal reparto della lavorazione immediatamente successiva ed, in definitiva, è la confezione che fissa i ritmi di tutte le fasi precedenti, fino all'acquisto delle materie prime. Tale sistema di pianificazione è noto con il termine giapponese "kanban" (cartellino segnaletico), di cui si parlerà in modo più approfondito in seguito;
10. **Responsabilità dinamica**: le responsabilità di reparto sono responsabilità di tipo statico, mentre le responsabilità dinamiche si riferiscono alle "responsabilità di flusso". Il sistema produttivo viene scomposto in *unità tecnologiche elementari* (*u.t.e.*) di cui fanno parte risorse umane della produzione, della pianificazione, della qualità e della

manutenzione. A ciascuna *u.t.e.* vengono delegate ampie responsabilità relative a tutte le operazioni che in essa vengono effettuate.

Il JIT, in quanto meccanismo che non tollera errori ed inefficienze, può presentare della problematiche: anche un breve ritardo da parte di un fornitore o di una lavorazione può comportare la paralisi dei raparti a valle. Per minimizzare questi rischi l'azienda deve aver creato al suo interno un ambiente di elevata qualità:

- nella ***progettazione*** e ***lavorazione***: i principi di razionalità e standardizzazione consentono grandi risparmi in termini di scorte di semilavorati in quanto i componenti modulari possono essere montati su più prodotti finiti;
- negli ***impianti***: essi devono avere la massima affidabilità in modo da ridurre al minimo possibile i tempi di fermo per guasto;
- nei ***sistemi informativi di produzione***: essi devono rilevare e comunicare in tempo reale l'avanzamento della lavorazione e la consistenza dei magazzini.

Per quanto riguarda l'ambiente esterno all'azienda, occorre prestare particolare attenzione a:

- ***fornitori***: per potersi approvvigionare, infatti, dei materiali solo quando sono effettivamente necessari alla produzione, bisogna scegliere fornitori *precisi ed affidabili*, che assicurino e garantiscano le consegne nelle scadenze e nelle quantità previste, nonché standard qualitativi elevati. L'azienda che intende implementare l'approccio JIT deve *fidelizzare il fornitore*, concentrandosi su pochi fornitori con i quali stipulare contratti aperti e di lunga durata; in questa ottica il fornitore viene visto come un *partner* [106], se non addirittura come un *alleato* [11] nella competizione con i concorrenti. Nel rapporto con i fornitori la forza contrattuale assume un ruolo importante: l'azienda "forte" potrà, ad esempio, imporre penali in caso di mancato rispetto delle consegne. Per poter lavorare JIT è necessario che tutti i fornitori siano in grado di lavorare secondo lo stesso modello. Non di rado ci sono aziende che adottano il *Just in Time Apparente*: in tale modello vi è un trasferimento della giacenza dell'assemblatore-produttore al

fornitore, poiché quest'ultimo continua a produrre secondo una logica *Make to Order*, o *Make to Stock*, ma spedisce secondo la logica JIT del cliente. In effetti quest'ultimo modello non fa che trasferire il costo di mantenimento scorta ad una fase più a monte della catena di approvvigionamento.

- ***trasporti***;
- ***ambiente sociale*** (effetti negativi di scioperi e assenteismo).

Infine, un aspetto molto importante è l'aspetto organizzativo: il JIT, in tal caso, prevede la riduzione dei livelli all'interno dell'organizzazione, che in questo modo assume una configurazione detta *lean*, ovvero *snella*. Il modello prevede, inoltre, la responsabilizzazione dei dipendenti a qualsiasi livello della piramide aziendale, nonché la stimolazione degli stessi ad evidenziare problemi e proporre soluzioni: in Toyota Motor Company tra il 1951 ed il 1989, il numero medio di suggerimenti per impiegato da 0,1 a 35; nello stesso periodo il tasso di adozione dei suggerimenti cresce dal 23% al 97% mentre nel 1992 ci sono stati 1.544.414 suggerimenti con il 99% degli stessi adottati [179]. Gli operatori, in particolare, devono adattarsi ai ritmi variabili imposti dal sistema: la flessibilità della produzione JIT si ottiene soprattutto attraverso la flessibilità della manodopera.
Nella Tabella 1.1 è stato effettuato un confronto tra il sistema JIT ed i sistemi tradizionali di gestione della produzione e delle scorte, per meglio evidenziarne le caratteristiche per le quali essi si differenziano:

**Tabella 1.1Il sistema JIT e i sistemi tradizionali a confronto**

| ELEMENTI | JUST IN TIME | SISTEMI TRADIZIONALI |
|---|---|---|
| **SCORTE** | Una passività da eliminare con ogni sforzo. | Una protezione contro errori, fermi e ritardi. |
| **LOTTI** | Fissati al minimo possibile sia i lotti di produzione che quelli di acquisto. | Fissati in modo da bilanciare i costi di giacenza ed i costi di set-up. |
| **ATTREZZAGGI (SET-UP)** | Fare in modo da renderli brevissimi così da produrre | Tempi di set-up poco considerati. |

| | | |
|---|---|---|
| | una grande varietà di parti. | |
| **FERMI MACCHINA** | Vanno eliminati, il problema deve essere risolto alla radice. | Inevitabili, fronteggiabili con scorte di semilavorati. |
| **FORNITORI** | Rapporti di fiducia lunghi e durevoli (fornitori come partner). | Rapporti precari. |
| **QUALITA'** | Zero difetti. | Tollerati alcuni scarti. |
| **LEAD TIME** | Da ridurre al minimo. | Da fronteggiare con le scorte. |
| **OPERAI** | Gestiti con il consenso. | Gestiti con norme e regole. |

### *1.2.2. Sistema Pull/Kanban*

Logica "Push" e "Pull"

Per meglio esplicitare il concetto di produzione "pull" e "push", è bene prima avere chiari in mente quelli di tempo totale di produzione e tempo di consegna. Il **tempo totale di produzione** $T_P$ viene definito come il tempo di attraversamento cumulativo di un prodotto, dal momento in cui vengono ordinate le materie prime a quello in cui esse vengono trasformate in prodotto finito, passando attraverso le varie fasi del processo. Esso rappresenta l'orizzonte temporale minimo con il quale la produzione deve guardare al mercato finale determinando la lunghezza del programma di produzione.

Il tempo totale di produzione è dato dalla somma di due lead time[2]: il lead time di produzione ed il lead time di approvvigionamento.

Il lead time di produzione è l'intervallo di tempo che intercorre dal momento in cui sono disponibili i prodotti in input a quando è disponibile il prodotto in output (il primo elemento del lotto); per misurarlo operativamente si potrebbe

[2] Letteralmente, "Tempo di Attraversamento".

marchiare il materiale in ingresso e cronometrare il tempo che impiega ad uscire dalla fase considerata. Il lead time di approvvigionamento, invece, viene definito come l'intervallo di tempo che intercorre dal momento in cui viene ordinata la merce a quando essa è disponibile per la produzione.
Quindi si ha:

$$T_P = LT_P + LT_A$$

Il **tempo di consegna** $T_D$ ( Delivery Time), invece, rappresenta l'intervallo di tempo compreso tra il momento in cui il cliente ordina un prodotto ed il momento in cui vuole che questo prodotto gli venga consegnato. Il suo valore è generalmente fissato dal cliente o dal mercato ed è, quindi, un dato non modificabile dalla produzione. Il $T_D$ dipende, ovviamente, dal tipo di business considerato; nel caso di produzione a magazzino, è dell'ordine di poche ore, mentre nei casi di produzione su commessa assume valori maggiori dello stesso tempo $T_P$.
Nella maggior parte dei casi $T_P$ è maggiore di $T_D$ e sono necessarie di conseguenza delle previsioni per approvvigionare i materiali e realizzare le operazioni produttive.
Se $T_P > T_D$ il programma di produzione si estende per un orizzonte temporale pari a $T_P$ si può riuscire a colmarlo di ordini di produzione solamente sino all'istante $T_D$; l'intervallo rimanente $T_P$-$T_D$ deve essere gestito tramite le previsioni. È importante considerare quello che tutto ciò significa dal punto di vista dell'investimento: un rapporto $T_P/T_D > 1$ implica la necessità di un investimento di capitale al momento $T_P$ con un ritorno previsto al momento $T_D$ (momento in cui termina la fase a rischio). Tale situazione è analoga ad una consueta decisione di investimento finanziario, quale l'acquisto di azioni o obbligazioni, in cui ci si chiede se il ritorno dell'investimento sia adeguato, considerati tutti i rischi connessi alla inaffidabilità delle previsioni, all'obsolescenza ed al deterioramento. Il rischio è tanto maggiore quanto più grande è l'intervallo $T_P$-$T_D$ e si comprende, dunque, l'importanza di minimizzarlo.

Nel caso in cui, invece, $T_P < T_D$ il programma di produzione è già totalmente definito dagli ordini che si estendono addirittura oltre il suo orizzonte temporale; nell'intervallo $T_D$-$T_P$ si possiede una certa libertà nella gestione delle priorità di soddisfacimento degli ordini, che si può sfruttare per ottenere una ottimizzazione delle fasi produttive. Il Lean Manufacturing System privilegia tale tipo di approccio.

In base a tutte queste considerazioni, un sistema viene definito:

- "Push", se $T_P/T_D > 1$;
- "Pull", se $T_P/T_D \leq 1$.

In un sistema di produzione di tipo ***"push"*** è necessario anticipare l'ingresso sia dei materiali in fabbrica, allo scopo di garantire il tempo di consegna richiesto dal mercato, sia degli ordini di lavorazione perché il tempo di attraversamento è più lungo dell'orizzonte del portafoglio ordini. Pertanto occorre far entrare in anticipo le materie prime e i semilavorati, producendo tramite previsioni di portafoglio ordini.

In questo tipo di sistema, quindi, i materiali vengono ***"spinti"*** secondo un piano prestabilito; di conseguenza, se le attività tra due stadi produttivi non sono ben coordinate, è inevitabile l'accumulo di scorte intermedie o di WIP (work in process) il cui effetto sarà quello di allungare il tempo totale di produzione invece di accorciare quello di consegna. Per comprendere meglio basti pensare, per esempio, ad uno stadio produttivo che continua a rifornire lo stadio a valle, che invece è bloccato a causa di un guasto. I sistemi di tipo "push" si basano solitamente su programmazioni tramite MRP.

In un sistema ***"pull"***, invece, i prodotti vengono ***"tirati"*** all'interno della produzione dagli ordini dei clienti e questi ultimi coprono ovviamente il tempo totale di produzione; circolando solo ciò che è necessario, in quanto si produce per soddisfare una precisa richiesta del cliente a valle, è possibile evitare l'accumulo di scorte intermedie, tanto inutile quanto oneroso. Tale logica è attuabile in realtà produttive in cui la domanda è poco variabile nel tempo. Sistemi "pull" puri sono molto rari nelle aziende manifatturiere, dove prevalgono, invece, situazioni in cui il portafoglio ordini viene completato da previsioni di vendita, almeno nella parte iniziale (sistemi "push-pull").

Un sistema pull, dunque, essendo interamente governato da ordini, sembra non necessitare di previsioni. Ciò, in realtà, è vero solo per i prodotti; occorre pianificare anche impianti e forza lavoro, risorse cioè che definiscono la capacità produttiva di un processo, in modo che anche queste siano approvvigionate con l'anticipo sufficiente a renderle disponibili al momento dell'utilizzo. Per entrambe queste risorse si potrebbero ripetere le considerazioni fatte nel caso dei materiali e valutare il rapporto tra il tempo di approvvigionamento ed il tempo di consegna; nella grande maggioranza dei casi esso risulta abbondantemente maggiore di 1, rendendo necessaria la loro pianificazione in base a previsioni.

I sistemi produttivi "pull" rappresentano allora un modello di eccellenza: essi costituiscono un target per quelli "push", raggiungibile attraverso l'abbattimento del $T_P$. tale operazione può essere effettuata, oltre che con strumenti quali l'ingegneria di prodotto e di processo, con interventi puramente gestionali. L'idea base muove dalla considerazione che il tempo di attraversamento aumenta al crescere del grado di integrazione verticale di un processo produttivo. Si può allora pensare di frammentare un sistema produttivo in *n* sottoinsiemi *(cellule)* indipendenti tra loro, ognuno caratterizzato da un tempo di attraversamento $T_P$ il cui valore sarà dell'ordine di $T_P/n$. Affinché un sistema a logica "pull" funzioni correttamente l'azienda deve disporre di un perfetto sistema di trasmissione delle informazioni lungo tutto il processo produttivo, in modo da sapere esattamente cosa produrre e quale ritmo sostenere.

Nell'azienda che utilizza un sistema "pull" i magazzini di materie prime e prodotti finiti praticamente non sono più necessari, mentre i magazzini di semilavorati lasciano il posto a piccoli polmoni: ogni centro di lavorazione è dotato di un punto di stoccaggio in uscita e di un punto di stoccaggio in entrata. Di seguito, nella Figura 1.1 e Figura 1.2, sono riportate le schematizzazioni dei layout, rispettivamente, di un'impresa "tradizionale" e di un impresa basata sulla logica "pull":

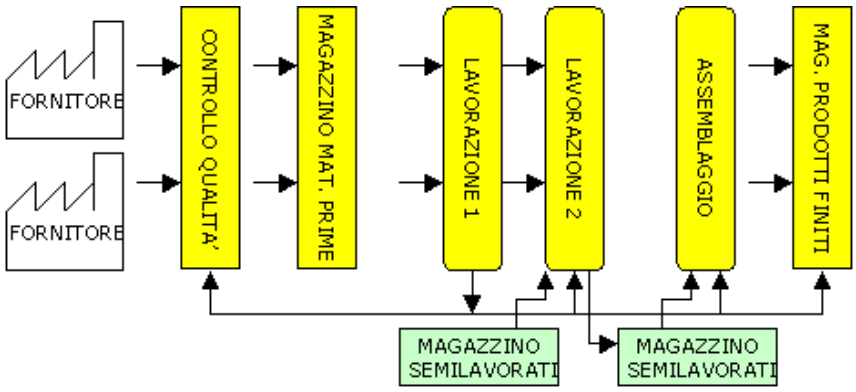



**Figura 1.1: Layout di un'impresa "tradizionale"**

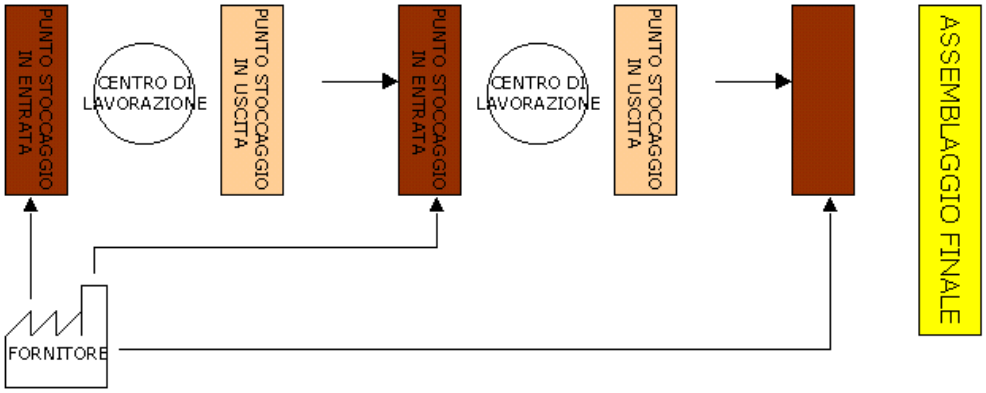



**Figura 1.2: Layout di un'impresa "pull"**

Un sistema di gestione di tipo "pull" ha il paradigma del suo funzionamento nelle considerazioni appena citate: esso crea, prima e dopo di ogni reparto produttivo, dei buffer di materiali di disaccoppiamento il cui scopo è quello di garantire il $T_D$ richiesto dal reparto immediatamente a valle. Ogni reparto della catena logistica vede, infatti, la valle come un cliente e il reparto a monte come un fornitore. Si noti che se ogni reparto deve produrre parecchi tipi differenti di pezzi, il livello totale di scorte può essere inaccettabilmente alto. In sistema di gestione basato completamente sull'approccio "push" viene meno questa visione segmentata del flusso produttivo per lasciare il posto ad un'ottica integrata di tutta la produzione ed, eventualmente, anche dell'approvvigionamento. L'eliminazione delle scorte è

un obiettivo dichiarato anche in questo secondo approccio, in cui un sistema di gestione centralizzato, tipo MRP, ha il compito di "spingere" i prodotti dentro la fabbrica e di regolarne l'avanzamento al suo interno.
Il principale inconveniente dei sistemi "push" è legato alle eventuali variazioni del piano di produzione: se esso cambia, i prodotti che sono stati già lavorati risultano non più necessari e devono quindi essere messi a magazzino in attesa di un loro eventuale futuro utilizzo.
Nei sistemi "pull", invece, il tutto inizia con l'ordine che tira la produzione di cellula in cellula, attraverso sistemi come il *kanban*, creando il minor numero di scorte di disaccoppiamento e permettendo, al tempo stesso, di lavorare per l'ottimizzazione dei tempi di attraversamento della singola cella.

*Il Kanban*

Il termine giapponese *"kanban"* significa, letteralmente, registrazione visiva o documentazione visibile ed indica una *"scheda"* o *"cartellino segnaletico"* che accompagna il singolo contenitore di materiale o di parti: non è necessariamente un cartellino fisico, in quanto può essere di tipo elettronico oppure rappresentato dal contenitore stesso.
Con il termine *kanban* si indica un sistema di programmazione, controllo e regolazione del circolante estremamente semplificato. Viene impiegato un sistema di schede che contengono le informazioni necessarie per la regolazione dei flussi dei materiali nelle diverse fasi della produzione; su di esse sono riportate le informazioni relative a cosa e quanto produrre o movimentare. Tali schede, circolando all'interno dello stabilimento tra i vari centri di lavorazione e stoccaggio, consentono una rapida ed efficace trasmissione di informazioni e permettono ad ogni centro di lavorazione di produrre solo ed esclusivamente ciò che verrà utilizzato a valle: in questo modo è possibile autoregolare il lavoro delle celle a fronte di variazioni del ritmo produttivo.
Il sistema di controllo kanban schematizzato in Figura 1.3 è evidentemente un sistema di controllo autoregolante. Se ad esempio la domanda di un componente è inferiore a quella programmata, il sistema è in grado di adattarsi da solo alle variazioni intervenute. Infatti, nell'ipotesi citata, non saranno più autorizzati

ordini di fabbricazione in numero superiore a quelli del numero di KP in circolazione.

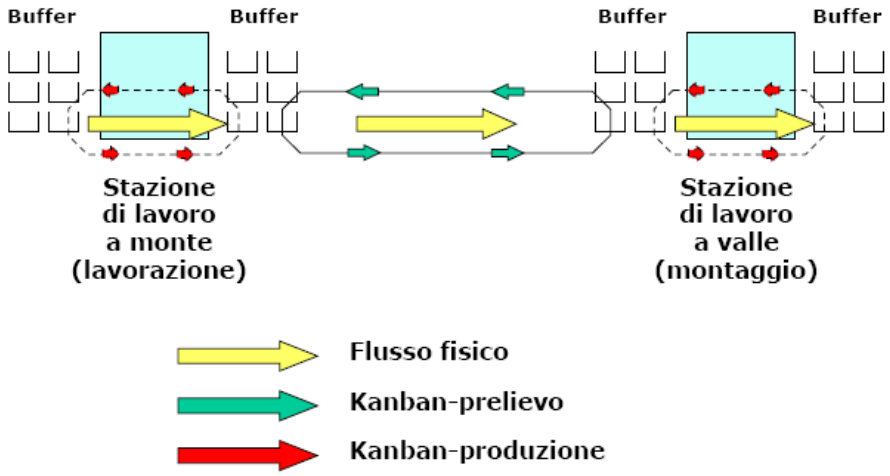



**Figura 1.3: Sistema di controllo kanban**

# 2.Stato dell'arte nei problemi di Scheduling della produzione

## 2.1. Introduzione

Ricerche sulla teoria dello scheduling sono state sviluppate da più di quaranta anni e sono state oggetto di molti studi che vanno da tecniche di dispatching rules grezze ad algoritmi altamente sofisticati di branch and bound paralleli ed euristici basati sul concetto di collo di bottiglia. Comunque con l'avvento delle nuove metodologie, come le reti neurali e l'evoluzione computazionale e con le ricerche provenienti da campi quali biologia, genetica e neurofisiologia, si è avuto un continuo contributo alle teorie di scheduling che enfatizza, appunto, la natura multidisciplinaria di questo campo. Uno dei più popolari modelli nella teoria dello scheduling è quello del job-shop scheduling che è considerato una buona rappresentazione di casi generali ed è noto per la sua elevata complessità di risoluzione. Esso è probabilmente il più studiato e l'avanzato sviluppo dei modelli di scheduling deterministico, lo portano ad essere utilizzato come campione comparativo per diverse soluzioni tecniche.

Formalmente, il problema di job-shop deterministico, verrà in seguito chiamato $\Pi_j$, consiste di un finito set $J$ di $n$ job $\{J_i\}_{i=1}^{n}$ processati su un set finito $M$ di $m$ macchine $\{M_k\}_{k=1}^{m}$. Ogni job $J_i$ deve essere processato su tutte le macchine ed è composto da un certo numero $m_i$ di operazioni $O_{i1}$, $O_{i2}$, ..., $O_{im_i}$ che devono essere schedulate in un dato ordine che viene chiamato *vincolo di precedenza.* Ci sono in totale $N$ operazioni, $N = \sum_{i=1}^{n} m_i \cdot O_{ik}$ è l'operazione del job $J_i$ che deve essere processato sulla macchina $M_k$ per un ininterrotto tempo di processamento $\tau_{ik}$. Ogni job ha un proprio flusso individuale sulle macchine che è indipendente dagli altri job. Inoltre, il problema è limitato da vincoli di capacità e da vincoli di disuguaglianza che stabiliscono che ogni macchina può processare solo una operazione e ogni operazione può essere processata su una sola macchina alla

volta. Se il tempo di completamento di $j_i$ sulla macchina $M_k$ è $C_{ik}$ allora il tempo finale dopo aver assegnato tutte le operazioni a tutti i job è detto *makespan* $C_{max}$. Nell'ottimizzazione del $\Pi_j$, l'obiettivo dello scheduling è quello di determinare il tempo di inizio di ogni operazione, $t_{ik} \geq 0$ , in modo tale da minimizzare il makespan soddisfacendo i vincoli di capacità e di precedenza. Quindi, l'obiettivo può essere espresso mediante la determinazione di $C^{'}_{\max}$, dove:

$$C^{*}_{\max} = \min(C_{\max}) = \min_{feasibleSchedules} (\max(t_{ik} + \tau_{ik}) : \forall J_i \in J, M_k \in M).$$

In una più generale affermazione dei problemi di job-shop, le replicazioni sulle macchine o l'assenza di macchine, sono allocate in un dato ordine del job $J_i \epsilon J$ , e così $m_i$ può essere più grande o più piccolo di $m$. L'attenzione principale è data al caso di $m_i=m$ non pre-emptive per il job $J_i$. Lo studio fatto è articolato come segue: le varie tecniche applicate al $\Pi_j$, si estendono dalle tecniche di approccio matematico e di branch and bound alle tecniche euristiche sui colli di bottiglia e si compara l'intelligenza artificiale alle tecniche di ricerca locale. Il metodo migliore è quello che rispetta i tempi computazionali e che ha un basso livello di divergenza dall'ottimo. Si analizzeranno, in seguito, anche possibili evoluzioni dei sistemi di scheduling.

Le tecniche di risoluzione possono essere dei metodi di *ottimizzazione* o di *approssimazione* e possono essere *costruttivi*, ossia costruiscono una soluzione a partire dai dati del problema, oppure *iterativi*, ossia modificano la soluzione con un continuo riordinamento della sequenza delle operazioni. Applicando un metodo approssimato si ottengono buone soluzioni in un tempo accettabile, invece, con procedure di ottimizzazione si producono soluzioni globali ottime, ma che richiedono tempi computazionali molto elevati. La maggior parte di questi metodi, che siano di ottimizzazione o di approssimazione, rappresentano il problema $\Pi_j$ usando il modello dei *grafi disgiuntivi*, $G=\{N, A, E\}$ di Roy e Sussmann (1964). Descriviamo prima il problema:

nel *grafo a nodi pesato* c'è un vertice per ogni operazione dove $N$ è il set dei nodi che rappresentano le operazioni che devono essere processate sul set di macchine

*M*. Inclusi tra il set N ci sono due speciali nodi fittizi, i nodi ◊ e ○, che corrispondono rispettivamente ai nodi iniziali e finali delle operazioni conosciuti anche come fonte e termine: *N={ ◊, 1, 2,…,○}*. Il peso, positivo, di ogni nodo j è equivalente al tempo di processamento corrispondente alla operazione, dove $\tau_{◊}$ $=\tau_{○}=0$ e il tempo di inizio e completamento di questi nodi rappresentano, rispettivamente, i tempi di inizio e fine di $\Pi_j$.

◊ è connesso all'operazione iniziale di ogni job e similarmente, le operazioni terminali sono connesse a ○.

In $\Pi_j$ ogni operazione *j*, eccetto ◊ e ○, ha esattamente due immediati predecessori e successori: sono il predecessore della macchina e del job *(JP(j) e mP(j))* e il successore della macchina e del job *(JS(j) e MS(j))*.

*A* è il set di archi connettivi direzionali rappresentanti i vincoli di precedenza per ogni job, tale che *(i,j)* Є A indica che l'operazione *i* è un immediato predecessore dell'operazione $j(i \prec j)$nel tipo di operazione del job. I vincoli di capacità sono rappresentati da uni-direzionali ma orientabili set di limiti, *E*, dove ogni membro di *E* è associato, tramite una coppia di archi disgiuntivi, che richiedono una macchina in comune, tale che *[i,j]* = $\{(i \prec j),(j \prec i)\}$ e $\{i, j \in O\}$. Uno scheduling è una soluzione possibile (P) del problema come segue:

min $t_{○}$ ○ЄN

subject to

| | |
|---|---|
| $t_i - t_j \geq \tau_j$ | (conjunctive constraint) $\forall i, j \in N(i, j) \in A$ |
| $t_j \geq 0$ | (earliest starting time constraint) $\forall i \in N$ |
| $t_i - t_j \geq \tau_j \quad \cup \quad t_j - t_i \geq \tau_i$ | (disjunctive constraint) $\forall i, j \in N i \neq j, (i, j) \in E_k, \forall k \in M$ |

Parametri importanti nella formulazione di tale grafo sono l'inizio e la fine delle operazioni con ordine fissato nella selezione corrente. L'istante di rilascio, $r_j$ , è la lunghezza del più lungo percorso, *l*, dalla sorgente all'inizio della operazione $O_j$, ed è dato da $r_j=l(◊,j)$. Analogamente, la lunghezza del percorso più lungo dal completamento di $O_j$ alla destinazione, chiamata $q_j$, ed è calcolata come $q_j=l(j,○)$.

Così se uno dei percorsi in $G$ va tra l'operazione $i$ e $j$, e $i \prec j$, allora $C_{max}=l(i,j)=r_i+\tau_i+\tau_j+q_j$.

## 2.2. Procedure di ottimizzazione

Nelle procedure esatte il tempo richiesto si incrementa esponenzialmente o come una polinomiale di alto grado per problemi lineari di dimensionamento, eccetto per delle versioni ristrette (casi speciali) di $\Pi_j$.
Un algoritmo efficiente risolve un dato problema ottimamente con una richiesta che si incrementa in maniera polinomiale in accordo con il dimensionamento degli input. Questi metodi, semplicemente, costruiscono una soluzione ottima del problema dato seguendo un semplice insieme di regole che determinano esattamente l'ordine di processamento.
Il primo esempio di un metodo efficiente ed il più semplice, probabilmente, lavoro della teoria dello scheduling è quello di Johnson (1956), il quale sviluppò un algoritmo, per un flow shop semplice con due macchine. che minimizza il massimo tempo di flusso. La notazione utilizzata per descrivere tale problema è $n/2/F/F_{max}$ , dove ogni job deve essere processato su tutte le macchine. Questo primo lavoro ha avuto una grande influenza sulle seguenti ricerche, perché il criterio di minimizzazione del makespan è stato attribuito a Johnson (1954). In aggiunta, questo algoritmo può essere semplicemente esteso per generare delle soluzioni ottime al problema $n/2/G/F_{max}$ e in casi speciali quale il problema $n/3/F/F_{max}$. Altri metodi efficienti sviluppati per il job shop sono quelli di Akers (1956) per problemi 2xm e Jackson (1956) per problemi nx 2, dove ci sono non più di 2 operazioni per job. Più recentemente Hefetz e Adiri (1982) hanno sviluppato un efficiente approccio per i problemi nx2, dove tutte le operazioni hanno un tempo di processamento unitario.
Comunque, il caso suddetto dei casi speciali di $\Pi_j$ è stato risolto ottimamente nei problemi di job shop con un numero di soluzioni possibili pari a $(n!)^m$. Un problema 20x10 ha $7.2651x10^{183}$ possibili soluzioni. Comunque molte di queste soluzioni non sono fattibili quando si considerano i vincoli di precedenza e

disuguaglianza, la completa enumerazione di tutte le sequenze fattibili per identificare la soluzione ottimale è non praticabile. A causa dell'elevata difficoltà computazionale, i problemi $\Pi_j$ sono considerati problemi decisionali di tipo NP. La notazione NP è data a problemi polinomiali non deterministici, ossia significa che non è possibile risolvere il problema in un tempo polinomiale P=NP(Cook 1971, Karp 1972, Garey e Johnson 1979). P è una sottoclasse di NP e consiste in un set di problemi che possono essere risolti deterministicamente in un tempo polinomiale ( tutti i casi speciali di suddetti $\Pi_j$, appartengono alla classe P). Un problema è NP-Completo se appartiene alla classe NP ed ha una difficoltà minore degli altri problemi di NP. La corrispondente ottimizzazione di tale problema è detta NP-Hard. Alcuni problemi NP-Hard possono essere risolti in un tempo polinomiale nel rispetto di differenti rappresentazioni degli inputs, ad esempio, nel caso di una macchina, un problema di total tardiness ha un tempo di calcolo polinomiale dato dalla somma dei tempi di processamento. Tali algoritmi sono conosciuti come pseudo-polinomiali. Comunque se un problema è descritto come strettamente NP-Hard o NP-Hard in senso stretto, tale che $\Pi_j$, allora una pseudo-polinomiale non può essere trovata per il problema se non in P=NP. Non sorprende che solo con una piccola variazione nella definizione del problema, quelli risolvibili efficientemente diventano rapidamente NP-Hard o strettamente NP-Hard. Per esempio, Lenstra et al.(1977) mostra che il problema 3x3, l'istanza nx2 con non più di tre operazioni per job e il problema nx3 con non più di due operazioni per job, sono tutte NP-Hard. Lenstra e Rinnooy Kan (1979) provano l'istanza nx2, dove le ultime operazioni per non più di due unità del tempo di processamento o il problema nx2, dove tutte le operazioni hanno un tempo di lavorazione unitario, appartengono entrambe all'insieme di istanze NP, anche se è permesso il pre-emption (Gonzales e Sahni 1978). Più recentemente, Sotskov (1991) prova la caratteristica NP nel problema 3xm, mentre Williamson et al. (1997) indicano che determinando l'esistenza di un programma con un limite superiore di quattro è NP-Completo anche quando tutte le operazioni hanno tempi di processamento unitario. In aggiunta, le soluzioni generate casualmente nelle relazioni di precedenza non sono distribuite uniformemente (Mattfeld et al. 1998) e, a differenza di altri problemi NP-Hard di ottimizzazione locale, non

portano ad un consolidamento delle caratteristiche soluzioni favorevoli. L'intrattabilità di $\Pi_j$ è maggiormente enfatizzata dal fatto che un problema proposto 10x10 da Fischer e Thompson (1963), potrebbe solo essere risolto da Carlier e Pinson (1989) sebbene sia stata provata l'implementazione di ogni possibile algoritmo. Se un algoritmo per $\Pi_j$ è processato in un tempo polinomiale, allora può solo garantire una soluzione che è ad una percentuale fissata, $\rho$, dall'ottimo. Tali metodi sono catalogati come algoritmi $\rho$-approssimati. Per esempio Shmoys et al.(1994) hanno proposto diverse approssimazioni poli-logaritmiche della schedulazione ottimale $\Pi_j$ che sono valutate in termini del loro errore relativo nel caso peggiore, mentre Fizzano et al. (1997) presentano una serie di algoritmi ad approssimazioni distribuite che configurano le macchine in una architettura circolare. Il più recente contributo in questa area (Williamson et al. 1997) prevede la prima non insignificante traccia teorica per indicare che i problemi di scheduling sono difficili da risolvere anche approssimativamente. Essi provano che per alcuni $\rho<5/4$ non esiste un algoritmo approssimato per $\Pi_j$ in un tempo polinomiale $\rho$, se non P=NP. Malgrado il progresso fatto in questi ultimi lavori, metodi efficienti non possono essere trovati per $\Pi_j$ dove $m\geq3$ e $n\geq3$ e French (1982) predice che algoritmi efficienti non saranno mai sviluppati per la maggior parte dei problemi di scheduling. Come risultato l'attenzione della ricerca di ottimizzazione è ritornata ad approcci enumerativi. Metodi enumerativi generano schedulazioni di una in una, usando procedure di eliminazione intelligente per verificare se la non ottimalità di una schedulazione implica la non ottimalità di molte altre che non sono ancora generate, evitando così la necessità di cercare uno spazio completo di fattibili soluzioni.

## 2.3. Formulazione matematica

### *2.3.1. Introduzione*

E' stato riconosciuto da molti ricercatori che i problemi di scheduling possono essere risolti in maniera ottimale, usando delle tecniche di programmazione matematica e una delle formulazioni più famose per il problema $\Pi_j$, è il programma intero lineare misto (MIP) formato da Manne (1960). L'MIP consta

semplicemente di un programma lineare con un insieme di vincoli lineari ed una singola funzione obiettivo lineare, ma con delle restrizioni aggiuntive di alcune variabili decisionali che devono essere intere *($y_{ipk}$)*. Qui, le variabili intere sono binarie e sono usate per implementare i vincoli disgiuntivi. *K* è un numero ampio e Van Hulle (1991) indica che per avere una regione di fattibilità occorre definire appropriatamente *K* che deve essere più grande di tutti, ma più piccolo del tempo di processamento.

Minimizzazione di $C_{max}$ soggetto a:

tempo di inizio $t_{ik} \geq 0$

$\{i, p\} \in J; \{k, h\} \in M$

vincoli precedenza $t_{ik} - t_{ih} \geq \tau_{ih}$ se $O_{ih}$ precede $O_{ik}$

vincoli disgiuntivi $t_{pk} - t_{ik} + K(1 - y_{ipk}) \geq \tau_{ik}$

$y_{ipk}=1$, se $O_{ik}$ precede $O_{pk}$

$t_{ik} - t_{pk} + K(y_{ipk}) \geq \tau_{pk}$

$y_{ipk}=0$, altrimenti

dove $K > \left( \sum_{i=1}^{n} \sum_{k=1}^{m} \tau_{ik} - \min(\tau_{ik}) \right)$

Malgrado l'eleganza concettuale, il numero di variabili intere scala esponenzialmente (Bowman 1959) e anche se sono usate le migliori e le più compatte formulazioni, esse, tuttavia, richiedono un numero elevato di vincoli(Manne 1960). Giffler e Thompson (1960) hanno anche detto che programmi lineari non conducono a metodi pratici di soluzione mentre French (1982) esprime l'osservazione che la formulazione di un programma intero di problemi di scheduling è, da un punto di vista computazionale, infattibile.

Nemhauser e Wolsey (1988) e Blazewicz et al. (1991) inoltre, enfatizzano tali difficoltà e indicano che i modelli di programmazione lineare non hanno conseguito un superamento per i problemi di scheduling. Come risultato queste tecniche sono solo buone per risolvere altamente semplificati problemi, entro un tempo ragionevole. Ciò, non sorprendentemente, suggerisce che tecniche idonee per $\Pi_j$ sono da trovarsi in altri contesti.

Alcuni successi, che sono stati conseguiti utilizzando la formulazione matematica, possono essere attribuiti agli approcci con rilassamento Lagrangiano LR (Fischer 1973a, b, Van De Velde 1991, Della Croce et al.1993 Hoitomt et al.1993) e i metodi di decomposizione (Ashour 1967, Applegate e Cook 1991, Chu et al. 1992, Kruger et al. 1995). Nei metodi LR vincoli di precedenza e capacità sono rilassati usando moltiplicatori lagrangiani non negativi, anche con termini di penalità incorporati nella funzione obiettivo, mente gli approcci a decomposizione dividono il problema originale in una serie di problemi minori, sotto-problemi più maneggevoli, che sono anche risolvibili in maniera ottimale.

I risultati indicano che ugualmente queste strategie soffrono di un eccessivo sforzo computazionale, mentre le soluzioni risultanti sono, solitamente, di qualità scadente, risultando molto distanti dall'ottimo. Anche quando queste formulazioni matematiche sono combinate con altre tecniche e applicate nel calcolo del limite inferiore (Fischer et al., Applegate e Cook), non generano buone prestazioni. I risultati mostrano che i limiti inferiori da essi generati non sono molto buoni e possono essere difficili da calcolare,inoltre, in alcuni casi, per colpa degli eccessivi tempi di calcolo richiesti per la ricerca, quest'ultima deve essere terminata prima.

E' evidente che l'approccio matematico è inadeguato per $\Pi_j$. Di conseguenza, l'attenzione principale per gli approcci enumerativi per il job shop è posta sulle tecniche di branch and bound.

### *2.3.2. Tecniche di Branch and Bound*

Gli algoritmi di branch and bound (BB), una struttura ad albero costruita dinamicamente, che rappresenta lo spazio di soluzione di tutte le sequenze fattibili. La ricerca inizia dal nodo più alto (root) e una selezione completa è conseguita una volta che il nodo del livello più basso (foglia) è stato valutato. Ogni nodo al livello $p$ nell'albero di ricerca, rappresenta una sequenza parziale di $p$ operazioni. Come implicato dal loro nome un branching, come pure uno schema di bounding, sono applicati per l'operazione di ricerca. Da un nodo non selezionato (attivo), l'operazione di branching determina il nuovo insieme di possibili nodi da cui la ricerca potrebbe progredire. Le due più comuni strategie di branching sono la Generating Active Schedules (GAS) e la Settling Essential Conflicts (SEC) (Lageweg et al. 1977, Barker e McMahon 1985). GAS è derivato dal lavoro di Giffler e Thompson (1960). Qui ogni nodo fa parte di una schedulazione parziale e il meccanismo di branching fissa l'insieme delle operazioni della nuova sequenza, mentre nel SEC il branching determina che $O_i$ potrebbe essere processato prima di $O_j$ o viceversa. Barker e McMahon (1985) indicano che il SEC prevede incrementi di flessibilità e in generale si è rivelato essere superiore al GAS.

Le procedure di ricerca selezionano le operazioni che continueranno la ricerca ed è basata su una stima di LB e sul migliore UB conseguito correntemente. In molte tecniche BB, la prima UB è normalmente prevista dalle medie di un euristica ed è applicata prima che la ricerca di BB attuale inizi. Se ad ogni nodo la stima LB si rivela essere più grande della migliore UB corrente, allora non c'è bisogno di continuare la ricerca e inoltre, con questa selezione parziale, non è possibile migliorare l'esistente UB. Di qui la selezione parziale e tutte le sotto sequenze sono disgregate. Una volta che un nodo foglia o un nodo dove LB è più grande della migliore soluzione corrente UB è analizzata, la ricerca ritorna (backtracks) al più alto nodo dell'albero non analizzato. La ricerca termina una volta che tutti i nodi sono stati implicitamente o esplicitamente ricercati.

Limiti rigidi sono perciò essenziali alle tecniche BB per impedire la necessità di ricercare sezioni molto ampie dello spazio decisionale. Molti tipi di limiti sono descritti nella letteratura. Sebbene Akers (1956), Brucker (1988) e Brucker e Jurisch (1993) hanno generato più LB riducendo $\Pi_j$ in sotto problemi di

dimensionalità *2xm, 2xm e 3xm* rispettivamente, la più popolare formulazione è la decomposizione dell'insieme di operazioni in *m* problemi da una macchina ($1|r_j|L_{max}$). Il limite alla macchina singola è ottenuto dal makespan del processo collo di bottiglia che è il limite più forte dopo tutte le macchine. Malgrado il fatto che questo problema è NP-Hard (Lenstra et al.), sono state sviluppate buone tecniche (Potts 1980, Carlier 1982).

In aggiunta alle strategie di branch and bound, indici di priorità o proposizioni, con le quali si cerca di fissare l'ordine di diverse operazioni, sono anche parte integrante di molti algoritmi BB. Dalla vincente combinazione questi tre componenti dello spazio di risoluzione possono essere rimossi dalla considerazione ad un primo stadio della ricerca. [So veda la ricerca di Pinson (1995) per maggiori dettagli.]

La ricerca tecnica BB fu inizialmente studiata da Brooks e White (1965), Ignall e Schrage (1965) e Lomnicki (1965). Altri lavori sono quelli di Brown e Lomnicki (1966) e Greenberg (1968). Balas (1969) presenta una delle prime applicazioni di un schema BB a $\Pi_j$. Questo metodo applica il modello del grafo disgiuntivo e considera solo le operazioni critiche. Un altro algoritmo BB per $\Pi_j$ è quello di Florian et al.(1971).

McMahon e Florian (1975) presentano una delle prime applicazioni positive della popolare decomposizione ad una macchina. Qui il branching comincia trovando il job critico, ad esempio quello che ha conseguito il massimo ritardo e determinando tutti i job con termini di consegna maggiore del job critico. Un altro algoritmo basato su principi molto simili è quello di Carlier (1982), dove l'algoritmo Schrage (Schrage 1970) è anche usato per generare una schedulazione iniziale. Un job critico C e un insieme critico J di operazioni, sono derivati da questa sequenza e la dicotomia definita dalla posizione di C relativa a J serve come base per il ruolo branching.

Seguendo questo lavoro Carlier e Pinson calcolano LB usando la schedulazione di Jackson Pre-emptive (JPS) basata sui più importanti lavori di Jackson. La macchina con il più ampio limite inferiore pre-emptive iniziale, che è determinato usando l'approccio di Carlier, è schedulata prima. L'insieme delle operazioni input (output) su questa macchina è utilizzato nella strategia di branching, la

quale fissa un'operazione non schedulata per farla eseguire prima (o dopo) una di quelle dell'insieme. Miglioramenti a questo lavoro sono stati apportati da Carlier e Pinson che hanno inoltre proposto due regole deduttive. In aggiunta una ricerca dicotomica è anche applicata per rafforzare l'iniziale LB. Nel loro più recente lavoro su $\Pi_j$ Carlier e Pinson hanno proposto quattro proposizioni e un altro schema limite inferiore per migliorare il branching e fissare le disgiunzioni.

Usando molte di queste regole applicate da Carlier e Pinson, Applegate e Cook determinano se una operazione non sequenziata *i*, potrebbe esserlo prima o dopo un fissato insieme di operazioni, *g*.

Questa strategia, chiamata *Ricerca del bordo*, determina a quale bordo di *g*, $O_i$ sarà schedulata ed è applicata nella strategia di un branching, così come in quella di bounding. Il ricercatore del bordo è anche stato applicato da Laurenço (1944), combinato con LBs, derivato decomponendo $\Pi_j$, in problemi di scheduling su una macchina con intervalli.

Un altro esempio di lavoro derivato da quello di Carlier e Pinson (1989), è l'approccio BB di Brucker et al. (1994) dove il branching è basato sullo schema a blocchi critico di Grabowski et al. (1986). Perregaard e Clausen (1995) modificano queste due tecniche in modo da permettere la ricerca dello spazio di soluzione più efficientemente. Il primo metodo prevede una strategia di ricerca parallela per l'algoritmo di Carlier e Pinson usando un approccio a carico bilanciato, mentre il secondo metodo è una versione parallela dell'algoritmo di Brucker et al. e applica un arrangiamento master/slave. Un altro esempio di tale algoritmo BB è quello di Boyd e Burlingame (1996) che costruiscono una versione parallela dell'algoritmo ricerca bordo. La prima strategia è applicata alla enumerazione parziale, l'albero è rappresentato da una collezione delle strutture dati mantenuta nella memoria condivisa.

Mentre tutti i metodi descritti sono così lontani dall'applicazione del modello a grafo disgiuntivo, Martin(1996) adotta una rappresentazione temporale orientata alla variante decisionale di $\Pi_j$. La più importante procedure di bounding creata è una tecnica chiamata shaving. Ogni operazione è allocata in una finestra temporale nella quale può essere processata e basandosi su varie regole e selezioni, uno o più tempi unitari, costituenti un tempo obiettivo $T$ , attendono di

essere rimossi (shaved) da *T*, riducendo la finestra temporale di varie operazioni. L'obiettivo è quello di ridurre, il più possibile, la finestra temporale di ogni operazione, evitando i conflitti tra le risorse.

*Analisi Comparativa*

Si noti che il confronto delle soluzioni e dei tempi per Perregaard e Clausen, e Boyd e Burlingame è relativo agli algoritmi che usano, rispettivamente 8 e 16 processori. Il risultato enfatizza che il risultato di McMahon e Florian è uno dei migliori metodi costruiti fino alla dimostrazione di ottimalità conseguita per FT 10.

Uno studio comparativo indica che il miglior metodo BB è quello di Martin. Comunque la tecnica shaving richiede tempi computazionali fenomenali e, in generale, il rendimento di queste tecniche BB è abbastanza sensibile a problemi individuali ed al limite superiore iniziale (Lawler et al. 1993).

Sebbene lo studio computazionale indichi che i miglioramenti sono stati ottenuti dai metodi BB, questo è principalmente attribuito alla tecnologia disponibile piuttosto che alle tecniche usate. In generale, queste non possono essere applicate ad ampi problemi e per eseguirle occorre conoscere molto bene il dominio di $\Pi_j$ con regole di inferenza altamente specializzate e le procedure di selezione sono richieste in modo da sondare i nodi ai livelli più alti nella soluzione ad albero, senza ricerche esplicite. Conseguentemente, le procedure ottimali sembrano inadatte per il job shop e molti ricercatori hanno volto la loro attenzione ai metodi approssimati.

## 2.4. Metodi approssimati

Sebbene i metodi approssimati non garantiscono l'acquisizione di soluzioni esatte, essi sono capaci di ottenere soluzioni vicine all'ottima, in tempi computazionali moderati e possono essere applicati con successo a problemi più ampi. L'importanza dei metodi approssimati è indicata da Glover e Greenberg

(1989) i quali suggeriscono che la ricerca per alberi direzionati è non soddisfacente per i problemi combinatori. I due indicarono che le euristiche ispirate dai fenomeni naturali e dalla risoluzione di problemi intelligenti, sono più adatte purché ci sia un collegamento bilaterale tra le operazioni di ricerca e di intelligenza artificiale. In questa analisi, sono considerate quattro categorie principali di tecniche di approssimazione:

- Regole di priorità
- Euristiche basate sul collo di bottiglia
- Intelligenza artificiale
- Metodi di ricerca locale

### *2.4.1. Regole di priorità (pdrs)*

Procedure approssimate applicate a $\Pi_j$ erano inizialmente sviluppate sulla base di regole di priorità e, data la loro semplicità di implementazione e la loro sostanziale riduzione dei tempi richiesti per il calcolo, sono le tecniche più conosciute (Baker 1974, French 1982, Morton e Pentico 1993).
Ad ogni step successivo a tutte le operazioni che sono disponibili per essere schedulate, è assegnata una priorità e l'operazione con la priorità più alta è scelta per essere processata. Solitamente sono processati secondo diversi pdrs in modo tale da conseguire risultati validi.
I più importanti lavori sul pdrs sono stati fatti da Jackson (1955, 1957), Smith (1956), Rowe e Jackson (1956), Giffler e Thompson (1960) e Gere (1966). Di particolare rilevanza è l'algoritmo di Giffler e Thompson. Questo è ora considerato come la base comune di tutti i pdrs e la sua importanza è derivata dal fatto che genera schedulazioni attive. La procedura comincia scegliendo le operazioni non sequenziate, $O_j$, con il tempo di completamento più vicino, poi trova tutte le altre operazioni che usano la stessa macchina ($M_k$) e con inizio più vicino al tempo di completamento di $O_j$. Queste operazioni sono piazzate in un insieme conflittuale.

Una operazione è poi selezionata da tale insieme e sequenziata il prima possibile. La procedura è ripetuta finchè tutte le operazioni non sono state poste nella sequenza. Le pdrs sono caratterizzate dal metodo applicato per selezionare le operazioni dall'insieme conflittuale, dove la regola più semplice per l'assegnamento è quella della scelta casuale.

Il più conosciuto e comprensivo studio di scheduling euristici è fatto da Panwalker e Iskander (1977) dove sono presentati, esaminati e classificati 113 pdrs. Blackstone et al. (1982), Haupt (1989) e Bhaskaran e Pinedo (1991), hanno procurato una estesa e sommaria discussione di questi ed altri pdrs. Una conclusione comune trovata in molti studi e, originariamente fatta da Jeremiah et al. (1964), è che per la misura della prestazione il makespan non domina le singole regole di priorità. Il più recente studio comparativo è quello di Chang et al. (1997) che valuta la prestazione di 42 pdrs, usando un modello di programmazione lineare. La loro analisi indica che le regole che considerano il più piccolo tempo di processamento (SPT) funzionano meglio rispetto a quelle che considerano il più lungo tempo di processamento (LPT). Poiché le regole individuali non sono efficaci e non procurano conclusioni chiare riguardo al criterio del makespan, hanno comportato lo sviluppo di più euristiche. Per esempio, l'algoritmo di Viviers (1983), incorpora tre livelli di classi di priorità per l'euristica SPT. Il metodo più comune di sviluppo di una soluzione è quello di avere una combinazione probabilistica delle pdrs individuali. I più importanti esempi di tale strategia sono dati da Crowston et al. (1963) e Fisher e Thompson. Lawrence confronta la prestazione di 10 regole di priorità individuali con una combinazione casuale di queste regole e mostra che i metodi combinati, procurano risultati di gran lunga superiori ma richiedono tempi computazionali maggiori. Altri metodi più sofisticati usati per controllare la scelta della regola di priorità da applicare includono un algoritmo genetico (Dorndorf e Pesch 1995) e la fuzzy logic (Grabot e Geneste 1994).

E' evidente che pdrs sceglie una possibile operazione da aggiungere alla sequenza parziale corrente, mentre le tecniche di branch and bound valutano tutte le possibili operazioni, implicitamente o esplicitamente. La tecnica di ricerca *a raggio* (Morton e Pentico 1993) fornisce un bilanciamento tra questi approcci,

valutando un numero di soluzioni migliori ed alcuni punti decisionali dati. Questo approccio è generalizzato dal ventaglio sequenziale di strategie a lista candidata (Glover e Laguna1997). Una strategia di ricerca a raggio filtrato è stata sovrapposta da Sabuncuoglu e Bayiz (1997) sulle decisioni selezionate fatte da un insieme di pdrs. Best descendants sono scelti, basati sul calcolo di LB, usando queste pdrs.

Un'altra tecnica che incorpora la ricerca a raggio, è l'algoritmo di inserzione job di Werner e Winkler (1995) che è composto da due fasi. La prima fase applica uno schema di inserzione per costruire una schedulazione ed stende l'algoritmo di Nawaz et al. (1983) per la permutazione del flow shop. E' posizionata nella schedulazione parziale una operazione tale che minimizza la lunghezza del percorso più lungo attraverso esso. La seconda fase applica una strategia di re-inserzione per migliorare interattivamente la soluzione iniziale dove i più vicini sono scelti sulla base di un approccio a blocchi critico. In entrambe le fasi, la ricerca a raggio è applicata per migliorare la ricerca.

*Analisi comparativa*

Un confronto tra i risultati di Sabuncuoglu e Bayiz e dieci differenti pdrs, che hanno prestazioni migliori rispetto al criterio di minimizzazione del makespan, è stato effettuato da Chang et al. Nella analisi è anche inclusa una regola che seleziona una operazione dall'insieme conflittuale a caso (random). In questo caso, tutte le operazioni hanno uguale probabilità di essere scelte e c'è una tecnica che casualmente combina tutte queste regole (combo). Data la natura di queste euristiche, ogni regola è stata processata dieci volte su ogni problema in modo da far diventare validi i risultati. Combo è stato applicato due volte. La prima applicazione impiega combo nello stesso modo di tutte le altre regole, per dieci processi ($combo_{10}$), mentre la seconda applicazione impiega combo come in Lawrence (1984), stoppando l'algoritmo se la migliore soluzione non è stata migliorata nelle ultime 200 ripetizioni ( $combo_{200}$). Tutti i legami sono rotti arbitrariamente.

Sebbene i risultati dei pdrs individuali sono conseguiti molto rapidamente, sono di qualità molto bassa (la deviazione dall'ottimo può essere al più del 74%) e in generale la qualità della soluzione degrada con l'incrementarsi della dimensione del problema. Questo è dovuto alla natura altamente miope di queste euristiche, poiché esse valutano solo una possibile operazione ad ogni punto decisionale, in tal modo considerano solo lo stato corrente della macchina e gli immediati dintorni. Poiché la singola regola non mostra una influenza chiara, è più prudente scegliere la migliore soluzione dalle diverse pdrs o applicare una combinazione di diverse regole. Però, questo richiede maggiori tempi di calcolo, come mostrato da $combo_{200}$ rispetto alle singole pdrs. I migliori risultati sono chiaramente quelli di Sabuncuoglu e Bayz, con un MRE medio di 8.33%, indicando come la ricerca a raggio sia appropriata e capace di migliorare le selezioni miopi, normalmente fatte da pdrs. Comunque, le deviazioni dall'ottimo sono tuttora alte, specialmente paragonandole con quelle degli altri approcci e in relazione ai tempi di calcolo rispetto alle pdrs individuali che sono di tre ordini di grandezze maggiori. I risultati suggeriscono che pdrs sono più utilizzabili come tecnica per la ricerca di una soluzione iniziale piuttosto che per considerare un sistema completo$\Pi_j$ e, per sfruttare veramente la ricerca a raggio, si potrebbe applicare in combinazione con altri metodi (Jain 1998).

### *2.4.2. Algoritmi euristici basati sul collo di bottiglia*

Sebbene per molti anni il solo metodo di approssimazione disponibile sia stato quello delle regole di priorità, recentemente, l'avvento di computer più potenti, così come l'enfasi posta sulla progettazione, analisi e implementazione tecnica (Fisher e Rinnooy Kan 1988), ha concesso approcci più sofisticati che permettono di eliminare il gap di tempo esistente tra le pdrs usate in maniera combinata e quelle usate nei metodi esatti. Un esempio di tale procedura è quello di Shifting Bottleneck Procedure (SBP) di Adam set al.(1988).

L'SBP è caratterizzato dai seguenti compiti: identificazione del sotto-problema, selezione dei colli di bottiglia, soluzione del sotto-problema e schedulazione della

nuova ottimizzazione. La strategia attuale si sviluppa rilassando il problema $\Pi_j$ in m problemi a macchina singola e risolvendo in maniera iterativa ogni sotto-problema $(1|r_j|L_{max})$ singolarmente, usando l'approccio di Carlier. Ogni soluzione a macchina singola è paragonata alle altre e le macchine sono classificate sulla base delle loro soluzioni. La macchina non selezionata avente il valore della soluzione più alto, è identificata come macchina collo di bottiglia. L'SBP mette nella sequenza le macchine collo di bottiglia basandosi su quelle già schedulate, ignorando le rimanenti macchine non inserite nella sequenza. La selezione della macchina collo di bottiglia è motivata dalla congettura che schedularlo ad uno stato successivo deteriorerebbe ulteriormente il makespan.

Ogni volta che la macchina identificata come collo di bottiglia è schedulata, tutte le macchine che lo sono state precedentemente, suscettibili di miglioramento, sono nuovamente ottimizzabili localmente risolvendo di nuovo il problema ad una macchina. Il maggior contributo di questo approccio è relazionare una macchina che permette di decidere l'ordine con cui le macchine potrebbero essere schedulate. Adams et al.(1988) si riferiscono a questa tecnica come SBI.

L'SBI è stato anche applicato ai nodi di un albero a enumerazione parziale (SBII), permettendo la considerazione di differenti sequenze di macchine. Una analisi numerica dei componenti individuali di SBP è procurata da Holtsclaw e Uzsoy (1996) e Demirkol et al. (1997) che indicano la qualità della soluzione e i tempi di calcolo come dipendenti significativamente dalla struttura del processo. Un esame comprensivo è previsto in Alvehus (1997).

Sulla base del SBII, Applegate e Cook hanno costruito una procedura per la soluzione iniziale conosciuta come "Bottle-k" (k è scelto come 4, 5 e 6) dove per le ultime k macchine non schedulate, l'algoritmo branch seleziona ogni macchina rimanente a turno. Un algoritmo chiamato "Shuffle" è anche stato formulato con la ricerca di bordo come nucleo. Da una schedulazione iniziale costruita tramite Bottle-k, Shuffle fissa l'ordine di processamento di uno o di un numero più piccolo di macchine, selezionate euristicamente con il resto delle macchine risolte ottimamente dalla ricerca di bordo.

Dauzère-Pérès e Lasserre (1993) e Balas et al. (1995) riportano diversi inconvenienti delle strategie proposte da Adams et al., che sono anche applicabili

all'SBP variante proposto da Applegate e Cook. Quando l'SBI produce la sequenza su di una macchina, si possono creare vincoli di precedenza tra le coppie di job su di una macchina non sequenziata. Questi vincoli, conosciuti come vincoli di precedenza ritardata (DPCs), insorgono in quanto, sequenziando una data macchina, si possono imporre delle condizioni sulla sequenza di altre macchine, del tipo che il job i deve precedere il job j da almeno uno specificato lasso di tempo. Data ai job questa dipendenza, quando il problema di scheduling ad una macchina è rilassato, esso è meno vincolato di quello che dovrebbe essere, da cui, la vera macchina collo di bottiglia non è selezionata, la migliore sequenza non è calcolata, la nuova ottimizzazione non garantisce un decremento monotono del makespan e la soluzione finale SBI può essere infattibile.
Come conseguenza, Dauzère-Pérès e Lasserre (1993) propongono una strategia euristica mentre Dauzère-Pérès e Lasserre (1995) e Balas et al. (1995) utilizzano uno schema esatto in accordo con DPCs. Il più recente lavoro (Balas e Vazacopoulos (1998) fissa una variabile per la procedura di ricerca locale (GL) in SBP. GL applica uno schema di interscambio basato su una struttura di vicinato locale che riversa più una disgiunzione alla volta.

*Analisi comparativa*

Comparando i risultati di vari metodi applicati a $\Pi_j$ si evince, chiaramente, che la strategia proposta da Balas e Vazacopoulos (1998) è superiore, ottenendo la migliore soluzioni in tutti i problemi testati. Questo approccio è anche utile per conseguire il miglior limite superiore per i problemi aperti ABZ 9. La debolezza primaria di questo algoritmo, comunque, è l'elevato sforzo di calcolo richiesto, rispetto agli altri metodi SBP, perché, per conseguire questi risultati, è stata applicata molte volte la procedura di ottimizzazione. In aggiunta, le migliori soluzioni sono state ottenute da una diversa impostazione di parametri differenti. Una difficoltà generale per gli approcci SBP è il livello di programmazione richiesto che è molto sofisticato e tutta la procedura deve essere completata prima che sia ottenuta una soluzione. Non c'è neanche un metodo disponibile per

decidere il dimensionamento del sotto problema e il numero di macchine da fissare. Sebbene Balas e Vazacopoulos suggeriscono diversi schemi elaborati di riottimizzazione, al momento non c'è una strategia che indichi come questi sono processati meglio e che risolva i problemi dove i job sono molto lavorati, quindi, l'SBP richiederà modifiche. Tuttavia, l'SBP è una buona procedura di progettazione analisi e implementazione ed è stata incorporata in molti altri lavori (Caveau e Laburthe 1995, Yamada e Nakano 1996, Vaessens 1996) che hanno migliorato il limite superiore ed inferiore di diversi problemi difficili. L'SBP è anche stato applicato a diverse generalizzazioni del problema $\Pi_j$. Per esempio, Morton (1990) estende l'SBP per progettare lo scheduling. Ovacik e Uzsoy (1992, 1996), applicano questa tecnica che permette, facilmente, di testare i semiconduttori. Ramudhin e Marier (1996) adattano la procedura dell'assemblaggio ad applicazioni open shop, mentre Ivens e Lambrecht (1996) estendono l'SBP dando una varietà di vincoli addizionali. La più recente generalizzazione è applicata al problema di job shop con deadlines (Balas et al.).

### *2.4.3. Intelligenza artificiale*

L'intelligenza artificiale (AI) è un sottocampo della scienza informatica relativa all'integrazione biologica e all'intelligenza del computer. Essa trae le origini fondamentali, dalla comprensione biologica e dall'uso di principi che trovano soluzione nella natura. Due metodologie principali sono qui analizzate: approccio a soddisfazione vincoli e metodo a rete neurale. Molte altre tecniche di AI, come i sistemi esperti, sono state applicate a $\Pi_j$ comunque il loro effetto è stato limitato.

#### *Soddisfazione vincoli*

Le tecniche di soddisfazione vincoli mirano a ridurre le dimensioni effettive dello spazio di ricerca tramite l'applicazione di vincoli, che restringono l'ordine in cui le variabili sono selezionate, e la sequenza, in cui i possibili valori sono assegnati ad

ogni variabile. Dopo che un valore è stato assegnato ad una variabile, ciascuna inconsistenza che sorge è rimossa. Il processo di rimozione dei valori inconsistenti è chiamato *analisi di consistenza*, mentre il metodo di disfare precedenti assegnazioni è chiamato backtracking. Una ricerca di backtrack fissa un ordine sulle variabili e determina anche un fissato ordine dei valori di ogni dominio. I problemi di soddisfazione vincoli (CSP) sono risolti se è specificata una allocazione completa di variabili che non viola i vincoli del problema. Sebbene considerato all'interno del dominio di AI, molti metodi di scheduling basati su vincoli, applicano un sistematico albero di ricerca e hanno collegamenti chiusi con gli algoritmi BB.

Esempi di precedenti sistemi di scheduling basati sui vincoli includono uno schema di scheduling di Fukumori (1980), un sistema logico di Bullers et al. (1980) e un metodo di pianificazione chiamato "Deviser" di Vere (1983). Più recentemente Fox (1987) ha progettato e costruito un modello euristico di ricerca interattiva vincolo-direzionata chiamata ISIS (Intelligent Scheduling Information System), nel quale i vincoli sono utilizzati per limitare, guidare ed analizzare il processo di scheduling. E' stata anche creata una versione migliore di questa tecnica, che incorpora scheduling paralleli (Ow, 1986).

Modifiche all'ISIS hanno condotto all'OPIS, sistemi di famiglie di planning/scheduling ( Smith et al. 1986, Owe e Smith 1988) ed a sistemi di scheduling CORTES (Fox e Sycara 1990). Pesch e Tetzlaff (1996) notarono diversi altri sistemi vincolo-direzionati: Soja (Lepade 1985), OPAL (Bensana et al. 1988) e FURNEX (Slotnick et al. 1992). Comunque Pesch e Tetzlaff indicarono che molti di questi metodi procurano solo una linea guida di alto livello per gli schedulatori umani e perciò sono di basso utilizzo pratico.

A partire dalla costruzione di tali casi industriali per i sistemi $\Pi_j$, molti lavori sono stati effettuati applicando procedure a soddisfazione vincoli per risolvere modelli standard $\Pi_j$. Uno dei più recenti esempi di tale lavoro è quello di Erschler et al. (1976) che determina $[(i \prec j),(j \prec i)]$ dato $M_i = M_j$. Ogni volta che una disgiunzione può essere fissata tutti i tempi sono propagati così che ulteriori disgiunzioni possono essere fissate.

La maggior parte dei più recenti lavori sui metodi di soddisfazione vincolo sono stati concentrati entro un piccolo gruppo di ricercatori. In uno dei loro più recenti lavori, Fox e Sadeh (1990) procurarono una serie di approcci a soddisfazione vincolo per problemi di scheduling sempre più difficili, mentre Sadeh (1989, 1991) costruisce uno schedulatore di $\Pi_j$ nel breve termine basato sul rilassamento e sulla due date. Ulteriori miglioramenti sono apportati da Sadeh et al.(1995) che implementano un approccio che implica lo scheduling di operazioni con predefinito tempo possibile di inizio al più presto/al più tardi. Sadeh e Fox applicano una struttura che assegna una probabilità soggettiva, basata sulla contesa delle risorse ad ogni operazione non schedulata. Il più recente lavoro è dato da Cheng e Smith (1997) che costruiscono una procedura detta Precedence Constraint Posting (PCP). PCP è anche stata estesa a Multi-PCP (MPCP) dove è applicata diverse volte in modo da migliorare i risultati.

Nuijten et al. (1993) costituiscono una strategia deterministica di soddisfazione vincoli che usa l'approccio di Sadeh come inizializzazione. Consistenza è conseguita utilizzando la considerazione logica derivata da Carlier e Pinson. Lo studio di Baptiste e Le Pape (1995) indica che il numero di backtracks diminuisce significativamente quando sono processate più propagazioni, il costo di aggiunta di un addizionale vincolo di propagazione è più che bilanciato dalla riduzione degli sforzi di ricerca e gli algoritmi che usano strettamente il metodo di ricerca-bordo funzionano meglio di quelli che non lo usano. Questi autori presentano anche una serie di vincoli basati sulla ottimizzazione degli algoritmi (Baptiste et al. 1995). Nujitene Aarts (1996) adattano molti di questi metodi ai problemi di job shop scheduling a capacità multipla($MC\Pi_j$).

Altri recenti lavori relativi al precedente sono fatti da Caseau e Laburthe (1994, 1995), Harvey e Ginsberg (1995) e Pesch e Tetzlaff (1994) introducono il concetto di intervalli di lavoro. Date due operazioni *i* e *j* (che possono essere le stesse), dove $M_i = M_j$, allora un intervallo di lavoro (TI) consiste di tutte le operazioni che possono essere processate tra il più vicino tempo di inizio di i e il più lontano tempo di completamento di j. Le proposizioni di ricerca-bordo di Applegate e Cook, sono allora applicate per determinare l'ordine di operazione. Caseau e Laburthe (1995) formulano un modello ibrido a due livelli. Il primo step permuta

le digiunzioni sui percorsi critici, mentre il secondo step fissa una piccola porzione delle soluzioni e poi applica l'algoritmo di ricerca-bordo con gli intervalli di lavoro per completare il resto dello scheduling.
Harvey e Ginsberg (1995) presentano un metodo chiamato Limited Discrepancy Search (LDS) che tenta di rimuovere decisioni sbagliate fatte nella ricerca precedente, ricercando sistematicamente, tutte le possibilità che differiscono dalla possibilità scelta in al più un piccolo numero di punti decisionali, o discrepanze. Pesch e Tetzlaff applicano un approccio a decomposizione e usano il metodo BB di Brucker et al. (1994) per risolvere ottimamente i problemi generati.
I più recenti metodi basati su vincoli (Nuijten e Le Pape) applicano vari metodi a propagazione ed euristiche a selezione di operazione in modo da determinare dinamicamente la schedulazione di una operazione prima o dopo.

*Analisi comparativa*

Confrontando i risultati conseguiti da alcune di queste tecniche di soddisfazione vincoli, appare che sono state conseguite, solo di recente, soluzioni di qualità adeguata. Sebbene Baptiste et al. (1995), Caseau e Laburthe, Nuijten e Aarts e Pesch e Teztlaff producono buoni risultati, richiedono ingenti sforzi. I risultati di Caseau e Laburthe suggeriscono l'incorporazione di altre euristiche nei metodi di soddisfazione vincolo per migliorare i risultati. Questo è ulteriormente messo in evidenza da Nuijten e Le Pape con la loro combinazione di tecniche esatte ed approssimate che danno prestazioni migliori. Comunque queste tecniche hanno molte somiglianze con il metodo BB infatti, anch'esse sono molto costose. Di conseguenza si può concludere che ulteriore ricerca deve essere ancora fatta se si vuole sviluppare un approccio a soddisfacimento vincoli per i problemi $\Pi_j$.
Un suggerimento è quello che, poichè i metodi analizzati sono abbastanza generici e possono facilmente esser applicati ad altri problemi di scheduling in modo da migliorare i risultati e ridurre i tempi di calcolo per esser competitivi in entrambe le categorie di migliori metodi $\Pi_j$, è necessario incorporare più informazioni specifiche del problema. In tale sistema, il metodo a soddisfacimento

vincoli può localizzare promettenti regioni dello spazio di ricerca, ottenendo una soluzione sub-ottimale ad una fissata percentuale dall'ottimo, che può esser sfruttata efficacemente ed efficientemente dal metodo dei problemi specifici.

### *2.4.4. Reti neurali (NNs)*

Nelle reti neurali le informazioni sono fornite attraverso una massiccia rete interconnessa di processi unitari paralleli. La loro semplicità, insieme alla loro capacità di lavorare con tempi di calcolo distribuiti, così come la loro propensione ad imparare e ad essere generalizzata, ha fatto si che la metodologia delle reti neurali sia molto utilizzata, permettendo di essere usata in molte applicazioni della vita reale (Zhang e Huang 1995). Cheung (1994) descrive alcune delle più importanti architetture delle reti neurali, applicate per risolvere i problemi di scheduling: reti di ricerca (Hopfield net), reti ad errore di connessione (Multi – Layer Perceptron), reti probabilistiche (Boltzmann machine), reti di competenza e reti auto-organizzate. Comunque l'applicazione delle reti neurali ai problemi $\Pi_j$, è stata principalmente implementata nei primi due di questi paradigmi. Una analisi e osservazione dell'applicazione delle reti neurali nello scheduling è anche prevista da Wang e Brunn (1995).

Le reti di ricerca come Hopfield nets sono reti auto-associative non lineari che hanno dinamiche inerenti la minimizzazione la funzione di energia del sistema o la funzione Lyapunov. In molti dei metodi basati su Hopfield, un modello matematico è applicato come mappa per $\Pi_j$ sulla rete neurale.

Le reti ad errore di connessione, d'altra parte, sono svolte su esempi che prendono la forma di una mappa $f$: $S \subset R^n \rightarrow R^m$, da alcuni sottoinsiemi limitati arbitrariamente $S$ di uno spazio euclideo n-dimensionale ad uno spazio euclideo m-dimensionale. Quando una attività campione è applicata alla rete, l'errore di connessione è corretto, tramite regole, con pesi, in accordo con la mappa fatta. Specificamente, l'attuale risposta della rete è sotto-tracciata dalla risposta

obiettivo desiderata per produrre il segnale di errore. I pesi sono aggiustati in modo che la risposta della rete attuale si muova verso la risposta desiderata

*Reti Hopfield*

La rete Hopfield (Hopfield e Tank 1985) domina le reti neurali basate su sistemi di scheduling e quando è applicata a $\Pi_j$, lo scopo è di minimizzare la funzione energetica, *E*, che è basata sul makespan, soggetta a vari vincoli di precedenza e di risorse. Se i vincoli sono violati, è prodotto un valore penalità che incrementa *E*. In uno dei lavori più recenti, Foo e Takefuji (1988) usano una codifica simile alla formulazione TSP di Hopfield e Tank (1985) per fare una mappa di $\Pi_j$ su di una matrice bidimensionale *mn* di *(mn+1)* neuroni. Poi al modello è applicato un processo di simulated annealing per evitare i problemi di convergenza locale. Un TSP con matrice basata sulla codifica è anche applicata da Hanada e Ohnishi (1993).

Per migliorare i loro metodi più recenti Foo e Takefuji (1988) costruirono una Integre Linear Programming Neural Network (ILPNN) formulando $\Pi_j$ come un problema MIP. La funzione energetica è rappresentata dalla somma dei tempi di inizio di tutti i job e le soluzioni sono ottenute processando programmi lineari ed interi (binari) aggiustandoli fino alla convergenza.

Van Hulle (1991) indica che l'approccio di Foo e Takefuji (1988) non garantisce soluzioni fattibili. Come conseguenza, traslano la formulazione MIP in una programmazione formato "goal" che è poi tracciato su di una rete neurale. Anche Willems e Rooda (1994) usano una formulazione MIP ma provano a superare alcune delle difficoltà maggiori riducendo gli spazi di ricerca del loro ILPNN attraverso l'effettuazione di calcoli preventivi. In modo da superare le limitazioni interfacciate da i sistemi ILPNN, Zhou et al. (1991) propongono la costruzione di Linear Programming Neural Network (LPNN). Questi evitano l'uso di funzioni energetiche quadratiche, implementando, invece, una funzione lineare. Cherkassky e Zhou (1992) paragonano questo modello neurale con tre pdrs usati comunemente. Il risultato mostra che, tranne in un problema, le reti neurali

hanno prestazioni migliori. Altre LPNN implementazioni, includono quelle di Chang e Nam (1993) e Gag e Shuchum (1994).
Altre strategie di codifica applicate alle reti Hopfield includono una rete digitale simulata da Satake et al. (1994) e una rete neurale tridimensionale fatta da Lo e Bavarian (1993).
Un ulteriore lavoro di scheduling basato sulle reti neurali è quello di Sabuncuoglu e Gurgun (1996) che hanno applicato una struttura 3D, che è aumentata con un processore esterno, che effettua una ricerca locale nella forma di un algoritmo soglia accettante. Il livello soglia è previsto al 10% riguardo all'accettazione di mosse di disimpegno. Comunque lo scambio può risultare in una soluzione non fattibile poiché non è ristretta alle sole operazioni critiche ( Van Laarhoven et al. 1992).

*Reti a propagazione back-error*

Uno dei più recenti studi in BEP scheduling, è quello di Remus (1990) che ha sviluppato diversi modelli neurali BEP, che sono paragonati con le regole di regressione lineare.
Chryssolouris et al. (1991) ha creato un'applicazione BEP per la progettazione di un sistema manifatturiero che determina il numero di risorse richieste in ogni stazione di lavoro di un job shop. Khaw et al. (1991) implementa un approccio BEP, che riceve dati variabili come gli ordini, la capacità disponibile, ed i tempi di set-up. La rete a tre stati di Dagli e Huggahalli (1991) consiste di uno stato input che riceve i dati che sono stati processati usando l'algoritmo di Lawler, uno stato vincolo di riconoscimento ed uno stato di recupero. Un modello BEP a quattro stati è stato sviluppato da Hoong et al. (1991) che fornisce un indice prioritario riguardante l'allocazione dei jobs alle risorse, mentre Cedimoglu (1993) applica un NN per simulare un job shop di assemblaggio e lo shop floor collegati insieme dalle scorte.
In molti di questi metodi citati il dato di input, usato per la formazione, è preso direttamente dallo shop floor. Un altro approccio popolare è formare l'NN usando i

risultati generati dal pdrs. Per esempio Watanabe et al. (1993) usano le regole slack, mentre Kim et al. (1995) formano un NN sfruttando i risultati conseguiti dalle regole Apparent Tardiness Cost (ATC) (Vepsalainen e Morton 1987) e le regole Apparent Tardiness Cost with set-ups (ATCS) (Lee et al. 1992).

Un'altra costruzione comune è combinare il modello BEP con altre tecniche. La più importante è un'integrazione con un sistema esperto. Uno degli esempi più vicini di tale struttura ibrida è quello di Raselo e Alptekin (1989,1990). Le loro reti neurali classificano regole di priorità e determinano coefficienti che poi inviano ad un sistema esperto in modo che lavori lo scheduling richiesto.

Il modello BEP nell'approccio di Dagli et al. (1991) riceve un dato in relazione agli stati del job e della macchina dal sistema esperto e classifica queste informazioni in gruppi di job. Questa classificazione è inviata al modello Hopfield che schedula i job in accordo a predeterminate priorità. Sim et al. (1994) fissano sedici reti neurali BEP in un sistema esperto. Ogni NN corrisponde ad uno sviluppo di funzioni di attivazione ed al fine di riconoscere il contributo individuale delle varie euristiche nello sviluppo di tali funzioni dell'attivazione.

Invece di un sistema esperto Dagli e Sittisathanchai (1995) combina una rete neurale BEP con un'algoritmo genetico (GNS). In GNS la rete BEP mappa la schedulazione costruita dall'algoritmo genetico al valore obiettivo multiplo ed è realizzato dalla prestazione di un sistema esperto.

Più recentemente Jain e Meeran (1998) indicano che il modello di BEP tradizionale soffre di insuccesso della formazione e di convergenza alla soluzione ottima locale quando è interfacciato con problemi che comportano il tracciare sulla mappa input-output. Un modello BEP modificato che incorpora impulsi, tassi di apprendimento e parametri movibili supera queste deficienze. La loro struttura codifica il problema in modo che la richiesta di neuroni scala linearmente permettendo di risolvere ampi problemi. Comunque i sistemi precedenti sono inadatti per risolvere problemi generici e forniscono un'adatta schematizzazione input-output, poiché le nuove schedulazioni possono solo essere identificate con successo, se soggette alla condizione che non devono variare più del 20% dalla formazione data.

*Analisi comparativa*

Molti dei metodi Hopfield sono stati implementati in hardware con tempi di esecuzione dipendenti dai componenti elettrici usati e conseguentemente i risultati sono prodotti istantaneamente. Poiché questi modelli sono codificati usando un modello matematico, essi soffrono la richiesta di un eccessivo numero di vincoli, variabili ed interconnessioni, quindi possono trattare solo piccoli problemi e, bloccati spesso in minimi locali, non garantiscono una soluzione ottimale. I diversi metodi sono applicabili a problemi di particolare dimensionalità. Se non sono applicati ai problemi per i quali sono stati progettati, potrebbero generare dei malfunzionamenti. I problemi associati ai modelli BEP riguardano una eccessiva richiesta di neuroni e la preparazione non ottimale acquisizione di dati da un sistema esperto, da sequenze di training esistenti o da pdrs. In aggiunta, come la maggior parte di BEP e NNs sono combinati con altri metodi di ottimizzazione, questa di solito non è realizzata dalla rete BEP ma invece lasciata nei sistemi ad altre tecniche, con le reti BEP che sono usate soprattutto per un reperimento veloce del database. Di conseguenza le reti neurali non sono considerate correntemente per essere competitive con le migliori euristiche per ciascuna classe dei problemi di ottimizzazione (Osman e Kelly, 1996).

### *2.4.5. Approcci misti*

Esempi di approcci misti AI, includono un modello parallelo stocastico distribuito da Lo e Hsu (1993) che ha molte somiglianze con la rete neurale stocastica di Hopfield. Un metodo "Vibrating Potential" basato sulle analogie con la fisica, statistica e dinamica (Yokoi et al. (1994) dove il potenziale fisico di ogni job è

osservato come l'interazione energetica tra operazioni su macchina e un metodo di ottimizzazione Ant System (AS) per $\Pi_j$ da Colorni et al. (1994). AS è una popolazione basata su di una tecnica di ottimizzazione stocastica sviluppata da Denebourg, Pasteels e Verhaeghe (1983). Questi modelli hanno il comportamento che ha una colonia di formiche nel ricercare il più piccolo percorso dalla loro sorgente di cibo. La formica è un semplice agente cooperativo la cui ricerca inizia ad essere veramente effettiva quando, lavorando collettivamente con molti altri agenti semplici, permette ad un minimo locale di essere superato. Le formiche sono casualmente posizionate sui nodi di un grafo disgiuntivo e seguono una procedura Monte Carlo combinata con un algoritmo euristico "greedy" per decidere lungo quali nodi adiacenti muoversi. I risultati che si ottengono sono generalmente scarsi mentre i tempi computazionali e lo sforzo richiesto sono elevati.

*Metodi di ricerca locale e meta-euristici*

Per derivare una soluzione algoritmica per un dato problema di ottimizzazione combinatorio *P*, dove *R* è un insieme di soluzioni fattibili di *P*, è spesso necessario definire le configurazioni, per esempio, un insieme finito di soluzioni, una funzione dei costi da ottimizzare e un meccanismo di generazione, che è una semplice prescrizione per generare una transizione da una configurazione all'altra tramite piccoli cambi. Tali metodi sono conosciuti come tecnica di ricerca locale (Aarts e Lenstra, 1997).

Nella ricerca locale il meccanismo di generazione crea un intorno per ogni configurazione. Un intorno, *N(x)*, è una funzione che definisce una transizione semplice da una soluzione *x* ad un'altra soluzione inducendo un cambio che tipicamente può essere visto come una piccola perturbazione. Ogni soluzione $x' \in N(x)$ può essere raggiunta direttamente da *x* tramite una singola trasformazione parziale predefinita di *x* chiamata *mossa*, e *x* si sposta a *x'*, quando una transizione è effettuata (il termine soluzione è concepito nel senso di un'entità che soddisfa certe richieste strutturali di un problema, ma in cui non

tutte sono fattibili). Una simmetria si assume sia mantenuta nella maggior parte delle ricerche di intorno quando *x'* è in un intorno di *x* se, e solo se, *x* è in un intorno di *x'*. L'obiettivo di queste strategie è perturbare progressivamente la corrente configurazione attraverso una successione di intorni in modo da dirigere la ricerca ad una soluzione migliore. Il miglioramento è ricercato ad ogni step da metodi ascendenti standard, o in qualche (possibile) numero più ampio di steps attraverso metodi più avanzati. Nei casi in cui le soluzioni potrebbero avere un convolgimento non fattibile, il miglioramento è spesso definito in relazione all'obiettivo modificato che penalizza tali infattibilità. In questi metodi la selezione di un intorno è dettata dai criteri di scelta. Diverse procedure di selezione comune includono:

- la scelta del primo intorno trovato a costo inferiore (questo è conosciuto come primo miglioramento ed è applicato in algoritmi soglia);
- la selezione del miglior intorno in un insieme di intorni (questo è conosciuto come il massimo miglioramento ed è applicato in algoritmi ad inserzione ed in procedure di spostamento dei colli di bottiglia);
- la scelta del migliore campione di un intorno (la tabu search adotta questo criterio provvedendo al miglioramento della soluzione corrente, e questa è la procedura di selezione applicata in varie ricerche approfondite e dagli algoritmi genetici);

Da una prospettiva generale, la soluzione a $\Pi_j$ può essere considerata come una collezione di decisioni locali relazionate con le nuove operazioni da schedulare. Dorndorf e Pesch (1995) suggeriscono che un lavoro potrebbe essere costruito pilotando queste decisioni locali attraverso il dominio di ricerca in modo da determinare una soluzione globale di alta qualità in un ragionevole lasso di tempo. In tali lavori le decisioni locali, fatte tramite problemi euristici specifici miopi, sono guidate attraverso soluzioni locali ottimali tramite meta-strategie fondamentali. Questo dà origine alla iterazione degli algoritmi di ricerca locale o meta-euristici, che combinano le proprietà dei problemi specifici di una ricerca

locale con le proprietà generiche del meta-risolutore, permettendo di conseguire buone soluzioni. Tecniche meta-euristiche sono le più recenti sviluppate nei metodi di ricerca approssimata per la risoluzione di problemi di ottimizzazione complessi (Osman e Kelly, 1996). $\Pi_j$ meta-euristici sono basati sulle strategie di vicinato sviluppate da Grabowski et al. (1986,1988), Van Laarhooven et al. (1988) e Nowicki e Smutnicki (1996).

Nel dettagliare le tecniche meta-euristiche individuali, alcuni dei risultati generali conseguiti nella ricerca locale riguardo alla questione della complessità e dell'analisi teorica e sperimentale, sono già stati descritti precedentemente. Uno dei lavori analitici più vicini è quello di Evans (1987) che ricerca la efficacia dei meccanismi generativi della ricerca locale dalla prospettiva dello spazio di stato di una intelligenza artificiale.

Vaessens et al. (1995) presentano una architettura che cattura più di uno schema proposto e ritengono che un metodo di ricerca locale multi-livello meriti maggiori analisi. Pirlot (1996) indica che pochi studi comparativi sono stati effettuati riguardo ai meta-risolutori come la Simulated Annealing (SA), tabu Search (TS) e Genetic Algoritms (Gas) e, da una analisi effettuata, i Gas appaiono essere i migliori di questi metodi, sia empiricamente che analiticamente. In un lavoro recente Mattfeld et al. (1998) analizza la struttura della fitness di $\Pi_j$ rispetto a come appare per un algoritmo di ricerca adattiva. Questi indicarono che le euristiche di ricerca adattiva, sono tecniche di ricerca sfruttabili per $\Pi_j$. Tutto quel che è richiesto è uno strumento di analisi effettiva.

In aggiunta a tali osservazioni analitiche, sono stati effettuati anche studi sperimentali. I più recenti (Vaessens et al. 1996) indicano che, nei loro test bed comparativi, nessuno dei metodi analizzati potrebbe conseguire un MRE totale minore del 2% entro un tempo di esecuzione di 100 secondi con un computer indipendente. Molti lavori in questo contesto sono svolti da Johnson et al. (1988) che definisce la classe di complessità *PLS* (ricerca locale in tempo polinomiale) e Yannakakis (1990, 1997) che costituisce un compito formale con attenzione alla complessità teorica degli algoritmi di ricerca locale, così come definisce, più chiaramente, la complessità associata ai problemi.

*Problemi basati su metodi space*

I problemi basati su metodi space sono euristici bi-livello che generano molte differenti soluzioni iniziali, usando rapide tecniche costruttive per problemi specifici, che sono poi sviluppate dalla ricerca locale.

*Ricerca in problemi space e in euristiche space*

Storer et al. (1992) credono che molti metodi affrontano solo il problema di come ricercare lo spazio di soluzione, ma non dove ricercare. Di conseguenza, definiscono la ricerca degli intorni in problemi space e in euristiche space, piuttosto che nelle più tradizionali solution space, che sono basate sulle operazioni di scambio. Entrambe le definizioni, applicano una rapida euristica base *h*, in modo da generare un appropriato intorno di soluzioni. Nel problema space, *h*, è applicato per perturbare la versione originale del problema, mentre nell'euristica space, la ricerca si sofferma sulla capacità di definire versioni parametrizzate di *h*. Un semplice algoritmo migliorativo è poi usato per effettuare la ricerca nell'intorno.
Storer et al. (1995) combinano la rapida euristica base generata in Storer et al. (1992), con diversi algoritmi di ricerca locale iterati. I risultati indicano che l'algoritmo genetico per i problemi space, è la tecnica migliore. I buoni risultati di un approccio genetico sono dovuti al fatto che è relativamente semplice da codificare in un euristico space, rispetto alla simulated annealing ed alla tabu search.

*Procedure d ricerca adattiva casuale greedy (GRASP)*

GRASP ( Greedy Randomised Adaptive Search Procedure) è un metodo basato sul problema space che consiste di una fase costruttiva e di una iterativa. Nella fase di costruzione la soluzione è, appunto, costruita un elemento alla volta. Tutti i

possibili elementi nuovi che possono essere scelti, sono ordinati in una lista che rispetta la funzione di priorità greedy. Un numero di elementi migliori è piazzato in una ristretta lista di candidati (RCL). La natura adattiva del GRASP è derivata dalla sua capacità di aggiornare i valori associati con tutti gli elementi ad ogni iterazione, basati sulla selezione appena fatta. Mentre la natura probabilistica di questi algoritmi è contenuta dalla selezione casuale di un elemento dalla lista RCL.

Nella fase iterativa, è applicata una procedura di ricerca locale, che successivamente ricompone la soluzione corrente con l'elemento migliore del vicinato. Questa fase si ferma quando non sono più trovati vicini migliori. Poi il processo ritorna alla fase di costruzione ed è costruita una nuova soluzione iniziale. L'algoritmo termina una volta che è raggiunta la soluzione desiderata oppure quando è raggiunto un predeterminato numero di iterazioni e si considera la migliore soluzione trovata. Resende (1997) presenta una applicazione di GRASP per $\Pi_j$. L'RCL è composto dalle operazioni che, quando sono sequenziate di nuovo, potrebbero portare al raggiungimento di un tempo di completamento più basso della sequenza parziale.

### *Analisi comparativa*

Gli algoritmi creati da Storer et al. (1992) sono stati applicati solo ad alcuni problemi generati e per alcuni dei problemi più difficili sono abbastanza mediocri. Questo è dovuto al fatto che un metodo migliorativo iterativo è usato come tecnica di ricerca. Tuttavia, applicando altri algoritmi di meta-ricerca Storer et al. (1995) sono stati capaci di ottenere buone soluzioni in lassi di tempo ragionevoli sia sui propri problemi che su quelli esterni. Sebbene GRASP è stato applicato con successo a diversi altri problemi NP-Completi, i limitati risultati disponibili per $\Pi_j$ sono scarsi.

### *2.4.6. Algoritmi soglia*

Uno dei gruppi più popolari di metodi di ricerca locale iterativa sono gli algoritmi soglia, che scelgono una nuova configurazione se la differenza di costo in un certo intorno è sotto una data soglia (*L*), per esempio $f(x') - f(x) < L$.

*Miglioramenti iterativi*

Il più semplice esempio di algoritmo soglia è quello di miglioramento iterativo, dove le soglie sono fissate a 0, perciò solo le configurazioni migliori sono accettate. Da una schedulazione iniziale generata casualmente, questi metodi dirigono la ricerca all'ottimo locale, dalla soluzione che deve risultare almeno buona o migliore, rispetto a tutte le altre del vicinato. Una volta raggiunto l'ottimo locale poiché non è accettata una mossa non migliorativa, rimane intrappolato. IM è la classe più semplice della tecnica di ricerca locale iterata, formando la base di altri metodi più elaborati. Mentre IM accetta la prima soluzione migliorativa nel suo intorno, una semplice variazione di questo, conosciuta come steepest descent, valuta tutte le mosse nel suo vicinato e seleziona quella che procura il maggior miglioramento.
Aarts et al. (1994) applicano un algoritmo migliorativo iterativo multi-start (MSIM) che termina quando è raggiunto il limite sul tempo totale di processamento. Questo limite è lo stesso per tutti gli algoritmi valutati nel loro studio computazionale. I loro metodi si paragonano al MSIM, simulated annealing (SA), threshold accepting (TA) e genetic local search (GLS) quando applicati con gli intorni di Van Laarhooven et al. (1992) e Matsuo et al. (1988). Il metodo MSIM inizia da una sequenza generata casualmente. L'algoritmo, allora, iterativamente migliora la soluzione corrente con la migliore del vicinato. Una volta che non ci sono più intorni
migliori, per esempio è ottenuto un ottimo locale, la ricerca ricomincia da un altro punto di inizio scelto casualmente, dal quale è trovato un ottimo locale. Questo

processo è ripetuto finché il criterio di terminazione non è soddisfatto ed è trovata la soluzione migliore (si noti la somiglianza con GRASP).

*Threshold accepting (TA)*

Negli algoritmi di threshold accepting (TA) (Dueck e Scheuer 1990) le soglie sono non negative. $L$ è inizialmente stabilito con valore alto, evitando di accettare mosse non migliorative, gradualmente decresce a zero così che solo le configurazioni migliori sono scelte. La sola applicazione di TA a $\Pi_j$ è stata quella di Aarts et al. (1994) che determina i valori dei parametri adatti empiricamente applicando gli intorni di Van Laarhooven et al. (1992) con 30 differenti sets di valori soglia ad un problema 10x10.
Due varianti di TA includono l'algoritmo di (Great Deluge (GDA) e il Record-to-Record Travel (RRT) (Dueck 1993). Anche questi metodi accettano soluzioni deboli purché superiori una certa soglia. La differenza di GDA e RRT con TAs è che essi dipendono solo da una appropriata selezione dei singoli parametri. Perciò questi metodi non sono stati applicati a $\Pi_j$.

*Ottimizzazione large step*

L'ottimizzazione large step, sviluppata da Martin, Otto e Felten (1982, 1992), è una fase duale del metodo di ottimizzazione, composta da ampi step e poi da piccoli step. Piccoli step sono più comunemente usati da algoritmi meta-euristici , mentre step ampi si riscontrano nella applicazione di tecniche di problemi specifici, permettendo il superamento di minimi locali.
Glover e Laguna (1997) indicano sottoclassi di diversa influenza e approcci a perturbazione per azionare sequenze ottime locali per ottenere soluzioni di buona qualità. Questa è una tecnica relativamente nuova con solo limitate applicazioni a $\Pi_j$.
Prove effettuate da Laurenço (1993, 1995) indicano che tra SA e IM, piccoli step sono processati meglio da SA e, comunque, SA li prende più lunghi di IM. Delle

tecniche a step ampi analizzate, i metodi migliori sono le implementazioni a due macchine casuali. Uno basato sul metodo specificato da Carlier (2rand –car) e l'altro applicando la Earliest Due Data Rule (2rand-edd).
Nel più recente lavoro (Laurenço e Zwijnenburg, 1996) piccoli step sono eseguiti usando la tabu search e ampi steps sono svolti usando la tecnica 2rand-car. Con un'iterazione basata sul criterio di terminazione, i risultati indicano che l'ottimizzazione di ampi steps con la tabu search, risulta migliore rispetto all'uso di una sola tabu search, che a sua volta è migliore di una ottimizzazione di un ampio step usando la simulated annealing. Una strategia basata sull'ottimizzazione di un ampio step è anche stata appicata da Brucker et al. (1996, 1997), però verso un più ampio dominio dei problemi di scheduling.

*Analisi comparativa*

L'algoritmo TA è sicuramente migliore del metodo IM, benché i risultati raggiunti da entrambi siano piuttosto scarsi, nonostante essi siano processati per lo stesso tempo come gli approcci GLS e SA.
Una delle ragioni per cui si verifica ciò, è che il metodo IM distrugge una caratteristica sostanziale dell'ottimizzazione (Mattfeld, 1996). La tecnica di inserzione di Werner e Winkler (1995) è migliore di TA specialmente quando è incorporata con la ricerca a raggio ma non è così efficiente come i metodi GLS e SA. Questo è particolarmente dovuto al fatto che, in alcune delle più difficili istanze, le tecniche SA e GLS sono eseguite tra le 2 e le 15 ore. Sebbene l'approccio SA comporti risultati eccellenti, 15 ore per risolvere problemi con non più di 300 operazioni appare eccessivo. Malgrado questo ampio tempo di esecuzione di tutti questi metodi, le migliori soluzioni, benché limitate, sono ottenute mediante l'algoritmo di ottimizzazione ad ampi steps di Laurenço (1995). Comunque esso richiede uno sforzo di calcolo elevato.

## 2.5. Simulated annealing

Nella simulated annealing (SA) le soglie sono positive e stocastiche. SA è una ricerca random che fu introdotta per l'analogia con la fisica statistica relativa alla minimizzazione di uno stato energetico di un metallo caldo. Esso è basato sulle proposte indipendenti di Kirkpatrick et al. (1983) e Cerny (1985) per vincolare i problemi di ottimizzazione.
In SA le configurazioni sono analoghe agli stati di un solido, mentre la funzione di costo $f$ e il parametro di controllo $e$, sono rispettivamente l'energia e la temperatura.
Un notevole contributo alle funzioni di intorno per $\Pi_j$ è stato dato da Van Laarhooven et al. (1992). Tale contributo consiste nell'effettuare solo quei movimenti che sono ottenuti tramite la inversione dell'ordine di processamento di una coppia adiacente di operazioni critiche, soggette alle condizioni che queste operazioni devono essere processate sulla stessa macchina. Un tale intorno è basato sui seguenti presupposti:

- se $x \in R$ è una soluzione fattibile, allora scambiando due operazioni critiche adiacenti che richiedono la stessa macchina non si può mai pervenire ad una soluzione impossibile;
- se lo scambio di due operazioni adiacenti non critiche porta ad una soluzione $x'$, allora il percorso critico in $x'$ non può essere più breve del percorso critico in $x$ (perché il percorso critico in x esiste ancora in $x'$);
- partendo da qualche soluzione fattibile $x$, esiste qualche sequenza di movimenti che ricercherà una soluzione ottimale $x^*$(conosciuta come proprietà connettiva).

Oltre a questi intorni, Van Laarhooven et al. (1992), costruiscono un generico metodo SA che applica una schedulazione annealing regolata dalla dimensione dell'intorno. Un'altra importante definizione dell'intorno è data da Matsuo et al. (1988) che prevedono che se due operazioni critiche adiacenti $i$ e $j$ stanno per essere scambiate, allora il movimento non risulterà mai essere immediatamente un miglioramento, se entrambi i metodi MP(i) e MP(j) sono su di un medesimo percorso critico.

Yamada et al. (1994) formulano un metodo chiamato "critical block simulated annealing" (CBSA) che contiene una struttura di intorno derivata dai blocchi critici in una struttura SA. I valori di temperatura iniziale e finale sono definiti in termini di una probabilità di accettazione generata da una proporzione e di un processo di reintensificazione che è applicato al fine di migliorare la ricerca.
Yamada e Nakano (1995, 1996) fanno riferimento al metodo CBS con un generatore attivo di schedulazioni di Giffler e Thompson (1960) e una versione iterativa di SBP chiamata "Bottle Repair"(BR).
BR è applicata quando una schedulazione scelta dall'intorno è rigettata dal metodo SA. Più recentemente, Kolonko (1998) indica che il metodo SA riguardo a $\Pi_j$, non è un processo convergente ed gli intorni standard di $\Pi_j$ non sono simmetrici. Basandosi su questi risultati egli presenta un metodo ibrido che consiste di un algoritmo genetico insieme ad uno schema SA.

*Analisi comparativa*

Sebbene il lavoro di Matsuo et al.(1988) e Van Laarhooven et al. (1992) è fondamentale agli approcci della ricerca locale per $\Pi_j$, l'analisi dei risultati di questi metodi mostra che essi sono piuttosto miseri. Solo quando altre tecniche sono incorporate, per esempio algoritmi genetici e SBP, si hanno soluzioni che migliorano qualitativamente, Yamada e Nakano (1996) e Kolonko (1998), capaci di ottenere buoni risultati. Comunque le principali deficienze degli approcci SA sono gli eccessivi tempi di calcolo necessari per raggiungere risultati soddisfacenti e il problema maggiore dipende dalla natura dell'algoritmo dove vari parametri devono essere selezionati attentamente. Le possibili ragioni, per cui questi metodi richiedono tempi di calcolo elevati, possono derivare dal fatto che molti processi devono essere svolti prima che siano ottenute buone soluzioni.
A causa della loro alta richiesta di tempi di calcolo, diverse strategie sono state proposte per provare e accelerare questi algoritmi.
Johnson et al.(1989) suggeriscono di rimpiazzare il calcolo della probabilità di accettazione con una tavola di approssimazione lookup. Szu e Hartely (1987)

hanno creato un metodo creato Fast Simulated Annealing (FSA) che consente di ottenere lunghi passi in avanti in modo da accelerare la convergenza. Petrson e Anderson (1987) propongono una variante del metodo SA chiamata Mean Field Algorithm, che è basato su Ising Spin Glasses. L'obiettivo è quello di ottimizzare l'energia magnetica nel sistema. Glover e Greenberg (1989), suggeriscono di fondere il metodo SA con quello AI al fine di migliorare le procedure di selezione del movimento che, essi credono, dovrebbero essere basate sul ragionamento umano piuttosto che su di un criterio di accettazione con movimenti più deboli con una probabilità decrescente. Uno di questi più recenti suggerimenti per il miglioramento è fornito da Saveh e Nakakuki (1996), che costruiscono il Focused Simulated Annealing (FoSA). FoSA identifica iterativamente le maggiori inefficienze nella soluzione corrente e tenta di rendere queste inefficienze più ovvie alla procedura di ricerca, tramite appositi coefficienti.

## 2.6. Algoritmi genetici ( GAs)

Un'altra generica tecnica di ottimizzazione modellata sull'osservazione dei processi naturali è quella degli algoritmi generici (Gas). GAs sono basati su un astratto modello dell'evoluzione naturale, tale che la qualità dei singoli dia luogo al raggiungimento di un livello più elevato compatibile con l'ambiente (vincoli del problema) (Holland 1975, Goldberg 1989). L'analisi presentata qui, è in modo approssimativo basata sulla classificazione data da Cheng et al. (1996) i quali suddividono tutti i progetti applicati negli ultimi anni in 9 categorie:

1 Operation based
2 Job based
3 Job pair relation based
4 Completion time based
5 Random keys
6 Preference list based
7 Priority rule based

8 Disjunctive graph based

9 Machine based

Questi progetti possono essere raggruppati in due approcci essenziali codificati, diretti o indiretti. L'approccio diretto codifica la schedulazione $\Pi_j$ come un cromosoma e gli operatori genetici sono usati per evolvere questi cromosomi in schedulazioni migliori. Le categorie da 1 a 5 sono esempi di questa strategia. Nell'approccio indiretto una sequenza di preferenze decisionali è codificata nel cromosoma, per esempio, inviando regole per incaricare job e gli operatori genetici sono utilizzati per migliorare l'ordine delle varie preferenze. Una schedulazione $\Pi_j$ è allora generata dalla sequenza di preferenze.

Le categorie da 6 a 9 sono esempi di questo metodo.

Il primo metodo GA applicato a $\Pi_j$ è di Davis (1985) ed è un esempio dell'approccio indiretto. La sua tecnica costruisce un ordine di preferenze delle operazioni per ciascuna macchina. Un tale approccio è ulteriormente esteso da Falkenauer e Bouffouix (1991) che codificano le operazioni per essere processate su una macchina come una lista di preferenze consistente di una serie di simboli. Anche Della Croce et al. (1995) adottano questa strategia di codificazione e il passaggio dell'operatore, ma con un metodo che guarda al futuro (look-ahead) in modo da generare schedulazioni attive.

Una delle più recenti liste di preferenze è data da Kobayashi et al. (1995) dove un cromosoma è una serie di simboli di lunghezza $n$ e ciascun simbolo identifica l'operazione per essere processata su una macchina. Tamaki e Nishikawa (1992) applicano una rappresentazione indiretta basata su un grafo disgiuntivo. Un cromosoma consiste in una serie binaria corrispondente ad una lista ordinata di preferenze di bordi disgiuntivi.

Uno dei primi approcci diretti è di Nakano e Yamada (1991) i quali creano una codifica binaria basata sulle precedenti relazioni di operazioni sulla stessa macchina. Una strategia chiamata Forcing è anche adottata per modificare un cromosoma se un'operazione può essere left-shifted. Yamada e Nakano (1992) sono capaci di apportare miglioramenti a questo lavoro definendo un operatore di

crossover come GA/GT basato su un algoritmo di Giffler e Thompson (1960). Qui il cromosoma è una lista ordinata di tempi di completamento delle operazioni. Uno dei più recenti approcci diretti, è quello di random keys (Bean 1994). Per $\Pi_j$ (Norman e Bean 1997) ciascun gene (un random key) consiste di due parti : un numero intero dall'insieme ( 1, 2, …, *m*) e una frazione generata casualmente da (0, 1). La parte intera del gene è l'assegnazione della macchina, mentre la parte in frazione, sistemata in ordine non decrescente, determina una sequenza di operazioni su ciascuna macchina.

Bierwirth (1995) crea un algoritmo genetico con una trasformazione generalizzata in modo da migliorare i metodi già esistenti. Un cromosoma rappresenta una trasformazione di jobs. In un altro lavoro Bierwirth et al. (1996) analizzano tre operatori crossover, che rispettivamente conservano la relativa posizione e l'assoluta trasformazione dell'ordine di operazioni. Più recentemente Shi (1997) applica una tecnica crossover che casualmente divide un esemplare arbitrariamente scelto in due sottoinsiemi, dai quali sono prodotti gli offspring. Malgrado l'abbondanza degli schemi elaborati ed applicati a $\Pi_j$ da GAs, i risultati a cui essi pervengono sono piuttosto scarsi. Diversi lavori indicano che GAs non sono ben adatti per la regolazione precisa delle strutture che sono chiuse dinanzi alle soluzioni ottimali (Dorndorf e Pesch 1995, Bierwirth 1995) poiché gli operatori crossover generalmente perdono la loro efficienza nel generare schedulazioni fattibili. Per superare alcuni di questi problemi è applicata una forma di Calcolo Evolutivo conosciuta come Genetic Local Search (GLS) che incorpora una ricerca locale di intorni nella strategia GA. Dentro la struttura GLS un soggetto concepito dagli operatori GA è usato come la soluzione iniziale per la ricerca successiva di intorno che perturba la prole alla più vicina sequenza ottimale. La local search è allora operata sulla generazione successiva tramite la ricombinazione genetica degli operatori.

La superiorità di GLS su GAs è evidenziata da Della Croce et al. (1994) che incorporano diverse strutture di intorno nel loro algoritmo genetico e provvedono a risultati migliori. Uno dei più conosciuti lavori GLS è di Dorndorf e Pesch i quali propongono due approcci per risolvere $\Pi_j$. Il primo metodo porta ad una combinazione probabilistica di 12 pdrs ed è chiamata P-GA mentre il secondo

approccio, riferito a SB-GA, controlla la selezione di nodi per SBII. Risultati indicano che SB-GA è superiore a P-GA. Una simile struttura evolutiva di cromosoma è applicata da Pesch (1993) che applica una strategia GLS per controllare i sottoproblemi della selezione nell'approccio di decomposizione.
Una ricerca dettagliata dell'applicazione delle tecniche evolutive a $\Pi_j$ è prevista da Mattfeld (1996). Basandosi su questa analisi sono creati tre algoritmi evolutivi GA1, GA2 e GA3. L'esucuzione di GA3 è superiore a GA2 che a sua volta è superiore a GA1. Questo avviene perché GA1 è un semplice algoritmo GLS ottenuto applicando il solo vicinato di Dell'Amico e Trubian (1993) mentre GA2 incorpora distanze tra gli individui tramite i valori medi spaziali delle popolazione, mentre GA3 prevede ad ulteriori miglioramenti attraverso l'incorporazione dell'eredità, che permette agli individui di adattarsi e cambiarsi in base ad i loro ambienti.
Yamada e Nakano (1995b) generano due schedulazioni genitore. Iniziando la ricerca dal primo genitore essi iterativamente sostituiscono una soluzione nella popolazione corrente con una sequenza migliorata parzialmente nei confronti del secondo genitore. Ulteriori miglioramenti a questi metodi sono stati fatti da Yamada e Nakano dove sono applicate una sostituzione stocastica dello schema che favorisce soluzioni che sono sconosciute al secondo genitore e una strategia critica di vicinato. Entrambi questi metodi sono basati sull'idea di percorso nella tabu search (Glover e Laguna 1997).

*Analisi comparativa*

La maggior parte degli approcci GA sembrano dare risultati poveri dovuti alle difficoltà che essi hanno con operatori crossover e la schedulazione codificata. I metodi GLS a prescindere dal fatto che provvedono ad ottenere soluzioni migliori rispetto alle tecniche GA, in generale raggiungono questi risultati con popolazioni più piccole, e sono più robusti. Sebbene, a causa dell' iterazione, essi richiedano più tempo, dovuto proprio alla ricerca di intorni. I migliori metodi GLS appaiono essere quelli integrati con altre tecniche, per esempio Mattfeld con le popolazione

spaziali e l'eredità e Yamada e Nakano con gli intorni critici e le varie forme di probabilità di accettazione. Comunque per ampi e difficili problemi si è concluso che anche i metodi GLS non possono raggiungere soluzioni ottimali in un tempo accettabile (Mattfeld 1996).

## 2.7. Tabu search (TS)

La tecnica di ottimizzazione iterativa globale Tabu Search (TS) ha origine dalle dottrine generali sul problema di una soluzione intelligente ed è derivata dai lavori di Glover (1977, 1986, 1989, 1990). Essa vieta (fa tabu) movimenti negli intorni aventi certi attributi, con lo scopo di guidare il processo di ricerca lontano dalle soluzioni che sembrano duplicare o assomigliare a soluzioni precedentemente ottenute. In ogni modo lo stato tabu di una mossa non è assoluto. Il criterio di ricerca permette ad una mossa tabu di essere selezionata se essa attiene ad un determinato livello di qualità.

Le funzioni di memoria media e lunga possono anche essere applicate per prevedere un esplorazione più ampia dello spazio di ricerca. Le strategie con termine medio o intermedio sono basate sulla modifica di regole al fine di incoraggiare i movimenti e le soluzioni storicamente trovate buone, dove questi schemi, usualmente, ritornano per parti attrattive della ricerca che intensificano la ricerca stessa nelle sue regioni. I metodi a lungo termine, diversificano la ricerca in aree non esplorate in precedenza. Spesso essi sono basati sulla modifica di regole per incorporare attributi nella soluzione che non sono frequentemente usati. Descrizioni più dettagliate si ritrovano in Glover e Laguna (1997).

Laguna et al. (1991, 1993) presentano alcuni dei primi approcci TS nello scheduling. Essi creano tre strategie di tabu search basate su semplici definizioni di movimento.

Barnes e Laguna (1993) propongono sei componenti primarie che permettono produzioni effettive di scheduling tramite TS. Essi sottolineano anche la necessità per gli schemi di medio e lungo termine di memoria di essere abbinati con una

ristretta struttura tabu search. Essi notano che in generale le inserzioni più che le procedure di scambio sono preferite poiché esse procurano un più alto grado di perturbazione e inoltre la sovrapposizione di TS ad altri algoritmi euristici prevede un fertile dominio per il lavoro futuro.

Hara (1995) presenta una tecnica conosciuta come Minimum Conflict Search (MCS), utilizzando restrizioni come quelle definite dalla struttura di memoria breve nella tabu search. Un minimum conflict è la condizione sufficiente minimale che non è ottimale. In $\Pi_j$ il percorso critico è indicato per essere il minimum conflict. Sun et al. (1995) creano un semplice approccio TS basato su manipulating active chains (ACM). Una catena attiva, è un blocco su un percorso critico.

Un notevole contributo è previsto da Taillard (1994) poiché egli incorpora una strategia che accelera la ricerca tramite il previsto bisogno di ricalcolare i tempi di inizio di tutte le operazioni in modo da determinare il costo del movimento. Comunque, questa strategia è effettiva solo quando le schedulazioni semi-attive sono permesse e poiché essa è incapace di dare un esatto makespan del movimento, essa può solo essere considerata come un strategia di giudizio veloce. Tuttavia Taillard incorpora questo schema, con gli intorni di Van Laarhooven et al. (1992), nel suo algoritmo TS. Un'implementazione parallela dell'algoritmo è prevista anche riguardo al calcolo dei percorsi più lunghi. Comunque egli conclude che non è sfruttabile per $\Pi_j$.

Dell'Amico e Trubian (1993) inseriscono un metodo con una nuova soluzione iniziale nel loro algoritmo TS, che utilizza una proprietà bi-direzionale. Sono formulate due strutture gli intorni. Il primo intorno considera la possibile inversione di un numero superiore a tre di archi coinvolgendo *i*, *MP(i) e MP(MP(i))* (e rispettivamente *MS(i) e MS(MS(i))*. Se l'inverso di ciascuno dei tre archi è tabu, tutta la mossa è proibita, ogni volta, invece, che una mossa è accettata tutti i componenti mossi sono inseriti nella lista tabu. La seconda struttura di intorno è basata su blocchi critici e in modo da accelerare la ricerca, Dell'amico e Trubian adattano la strategia di stima di Taillard (1994) ai loro intorni.

Correntemente il miglior metodo TS, rispetto alla qualità della soluzione e al tempo, è quello di Nowicki e Smutnicki (1996) che applicano un intorno

altamente vincolato, che divide un singolo percorso critico in blocchi. Se ci sono *b* blocchi nell'intorno, allora per i blocchi $1 < l < b$, le prime e le ultime due operazioni sono scambiate. Mentre per il blocco $l=1$ $(l=b)$ solo l'ultima (la prima) dei due blocchi di operazioni è scambiata. La soluzione iniziale è generata dal metodo costruttivo ad inserzione proposto da Werner e Winkler (1995). La permanenza nella lista tabu è fissata ed è applicata una strategia di recupero delle soluzioni dove un numero di schedulazioni privilegiate sono fissate in una lista. E' importante sottolineare che i tempi di calcolo della CPU non includono i tempi della soluzione iniziale. Poiché la soluzione iniziale ha complessità $O(n^3m^5)$, Aarts (1996) dice che sono i tempi di calcolo, qualche volta, a sviare.

Aarts (1996) e Ten Eikeelder et al. (1997) costruiscono diversi algoritmi sequenziali e paralleli per $\Pi_j$. Questi divisero una tecnica di valutazione parziale conosciuta come strategie di stima accelerata di Taillard (1994) del (35-40)%. Il loro algoritmo TS è come quello di Nowicki e Smutnicki (1996) eccetto che per il mantenimento delle soluzioni tabu che sono cambiate casualmente. Analisi suggeriscono che non più di 20-30 processori paralleli possono essere usati nell'implementazione dell'algoritmo. Ci sono anche problemi che hanno mosse che non possono essere effettuate e quindi sono fatte tabu.

Uno dei più recenti approcci ( Thomsen 1997) è stato utile per migliorare il limite superiore di diversi problemi combinando la tabu search con un algoritmo di branch and bound. La strategia BB è usata per la diversificazione in modo da minimizzare ragionevolmente i tempi di calcolo è processata solo euristicamente per un numero ristretto di iterazioni e entro un limite di approfondimento dell'albero. L'intorno di Nowicki e Smutnicki (1996) è usato come schema di branching.

Diversi approcci di tabu search sono stati applicati a generalizzazioni del problema $\Pi_j$. Esempi sono quelli di Hurink et al. (1994) per macchine multi-purpose, Brucker e Kramer (1995), Daezère-Pérès e Palli (1997) e Baar et al. (1997) per i problemi di scheduling progettati con vincoli sulle risorse.

*Analisi comparativa*

In generale, TS procura i migliori risultati tra tutte le tecniche, dove ogni approccio TS è utile per generare buone soluzioni in tempi di calcolo ragionevoli. I risultati più deboli sono quelli di Sun et al. (1995) e ciò è dovuto al fatto che ha solo una memoria di breve termine. Nowicki e Smutnicki (1996) e Thomsen (1997) ottengono i risultati migliori, con tempi di calcolo ridotti e veloci. E' da notare la grande adattabilità dei modelli ibridi che combinano la tabu search con altre tecniche. Però la tabu search, come la maggior parte de i metodi di ricerca locale, richiedono la conoscenza di molti parametri e un aggiustamento specifico per ogni problema per essere implementati con successo e questo compito è difficile da svolgere.

## 2.8. Event-drive and time-drive simulation

### *2.8.1. I metodi di simulazione*

La simulazione può essere impiegata sia per sistemi discreti che continui, tuttavia l'analisi sarà effettuata solo in relazione alla prima tipologia di sistemi perché meglio rappresentativi dei sistemi di produzione. I metodi di simulazione per tali sistemi sono numerosi e possono essere classificati come mostrato in Figura 2.1.[150]

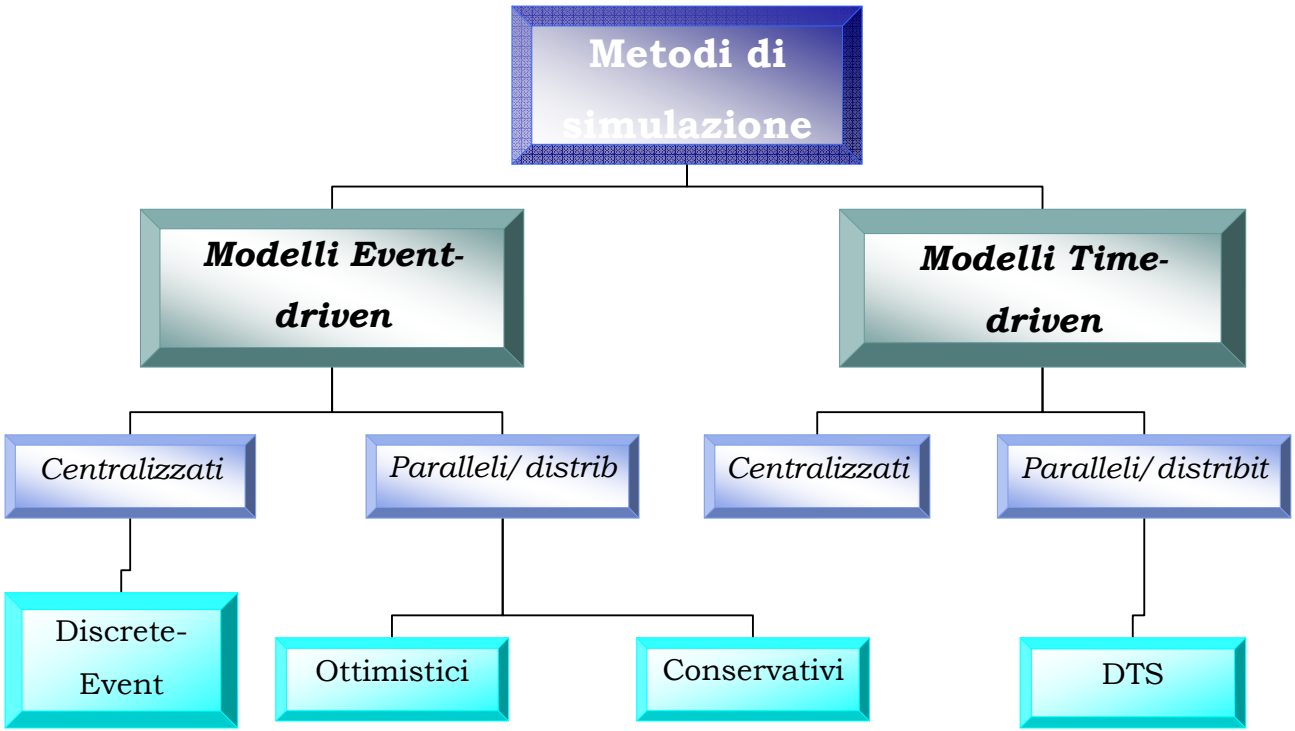



**Figura 2.1 Classificazione dei metodi di simulazione**

In generale, le simulazioni Event-driven sono caratterizzate dall'evoluzione del sistema in base all'accadimento di determinati eventi, quindi il clock della simulazione avanza in maniera discontinua. Invece, nel caso time-driven la schedulazione degli eventi è realizzata in base agli istanti di tempo, ovvero il clock di simulazione aumenta con un tasso costante. I sistemi di produzione possono essere realizzati impiegando entrambi gli approcci. Nel primo caso, si segue il percorso effettuato dalle parti, mentre nel secondo si osserva il sistema a diversi istanti di tempo. Entrambi le tecniche prevedono, comunque, che ad ogni osservazione lo stato di tutti i componenti del modello si modifichi. In genere, lo studio dei sistemi e i software commerciali sviluppati sono focalizzati sulla Discrete-Event Simulation (DES). Per questo motivo, gran parte degli approcci trattati impiegano tale tecnica. Tuttavia, è necessario considerare che numerose applicazioni recenti sono state sviluppate impiegando la Distributed Discrete-Event Simulation, che scompone problemi complessi su più processori e agli approcci di tipo time-driven centralizzati e distribuiti. Le distributed discrete-event simulations possono essere classificate in due categorie, metodi conservativi e ottimistici che si distinguono sulle modalità di evitare l'errore di casualità, ovvero l'errore conseguente al processamento di un evento prima del tempo minimo consentito. Ad esempio, Holthaus et al. [77] presentano una nuova metodologia per la schedulazione e la coordinazione di sistemi di produzione job shop decentralizzati basata sull'impiego di sistemi multi-computer. I sottosistemi paralleli del sistema integrato di produzione sono distribuiti attraverso una rete di computer. Gli esperimenti di simulazione mostrano performance significativamente migliori rispetto a quelle ottenute con le regole convenzionali di scheduling e tempi di run molto ridotti. In particolare, il modello sviluppato è impiegato dagli autori per la schedulazione di un sistema dinamico di tipo job shop permettendo il miglioramento simultaneo delle due date e del tempo di flusso.. Il modello è costituito da un insieme di processori di simulazione e da un controllore centrale connessi attraverso una particolare rete di comunicazione.

Naturalmente la velocità di simulazione dipende dal numero di processori e di sottosistemi allocati su un processore. In ogni caso, può essere individuato, sulla base di tempi critici di simulazione, in funzione della dimensione del sistema di produzione il range ideale del numero di processori che minimizza il tempo di esecuzione. Alcuni autori tra cui Shen [154] hanno proposto quale prospettiva futura l'implementazione di tali applicazioni distribuite sul Web. In particolare, gli autori descrivono un meccanismo di realizzazione della DES sul Web integrando tecnologie CORBA e Java, la prima come interfaccia per il Web e la seconda per lo sviluppo dei sistemi di simulazione.
La Distributed Time-Driven Simulation può essere ulteriormente migliorata impiegando un approccio time-scaling che permette un maggiore impiego della banda di rete e della CPU. Un esempio è presentato da Chao[45]. L'autore valuta l'impiego delle simulazioni time-scaled. distribuite (DTS) per problemi di schedulazione job shop con due parti su due macchine e sei parti su quattro macchine. I risultati ottenuti sono stati confrontati con quelli ricavati dall'impiego di un metodo di simulazione centralizzato che utilizza Arena RT . Dal confronto si evince che i tempi di simulazione sono molto ridotti rispetto al caso centralizzato; tuttavia tali metodi sono ancora limitati dalla dipendenza dalla capacità della rete e dai costi di sviluppo sostenuti. Per questo motivo, gran parte degli studi relativi alla pianificazione sono focalizzati su un tipo di approccio DES.

### *2.8.2. I modelli di simulazione come strumento di supporto decisionale*

In passato, la simulazione di sistemi di produzione e logistici è stata ampiamente analizzata in relazione a problemi strategici e tattici. La simulazione era impiegata per analizzare le cause di eventuali disastri nel sistema di produzione o come strumento per testare nuovi progetti di sistema prima di investire in essi.[90] Negli anni '80 software commerciali di

simulazione sono stati sviluppati e impiegati dalle grandi imprese, ma è stato comunque come uno strumenti di supporto tattico e strategico.

Un ambito interessante della simulazione è la simulazione operativa, sviluppata negli ultimi anni, in termini di supporto alla pianificazione e controllo di breve termine per sistemi di produzione e logistici. Questo tipo di simulazione presume la realizzazione di modelli di simulazione molto dettagliati e continuamente aggiornati in accordo con il sistema reale in tempi e costi ridotti. In particolare, grazie a questo tipo di modelli e all'integrazione con altri sistemi informativi è possibile realizzare la simulazione in real time o in parallelo con il sistema reale. Inoltre, accelerando il tempo di simulazione, l'analista può valutare le diverse alternative decisionali. I tipi di problemi in cui la simulazione operativa può essere usata è lo scheduling, la pianificazione della capacità e controllo.[45]

L'analisi dei possibili impieghi di questo tipo di simulazione nell'ambito dei processi produttivi sarà effettuata secondo lo schema logico rappresentato in

Figura 2.2

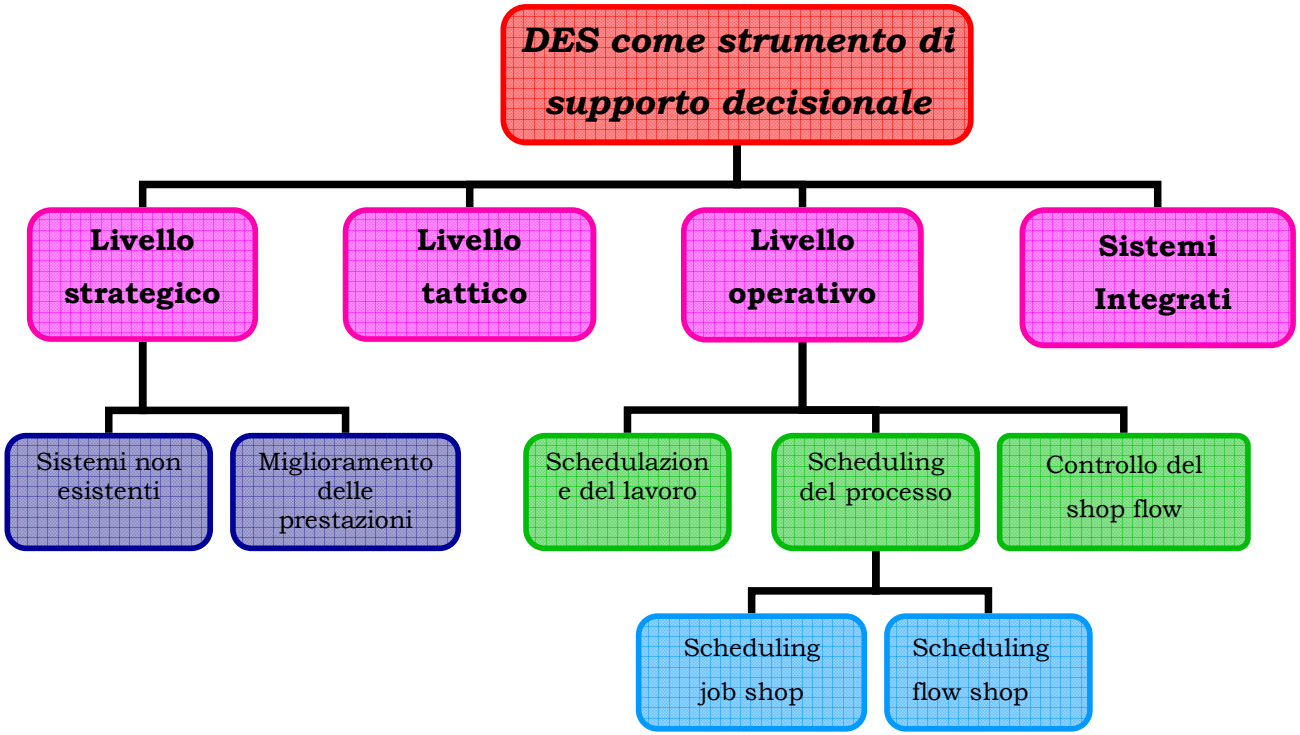



**Figura 2.2 Classificazione dei modelli di simulazione**

### *2.8.3. L'impiego dei modelli di simulazione a livello strategico*

Nell'ambito di impiego della simulazione come strumento di supporto tattico e strategico molti sono gli approcci impiegati. In genere, la simulazione è impiegata al fine di selezionare la migliore configurazione tra diversi sistemi. In particolare, si analizzano quattro diverse modalità di sviluppo dei sistemi di simulazione come strumento di supporto decisionale. Il primo caso concerne lo sviluppo di sistemi non esistenti. Gli altri tre approcci sono relativi all'analisi di sistemi di produzione esistenti.

*Lo sviluppo di modelli di simulazione per la progettazione di sistemi*

Un approccio basato sull'analisi di sistemi non esistenti è presentato da Ceric et al. [35] in cui la simulazione è impiegata sia per valutare l'influenza dei fattori sul sistema da realizzare e, quindi, selezionare le configurazioni da valutare sia nella fase di scelta tra le diverse alternative. In particolare, gli autori analizzano l'impiego della simulazione nell'ambito dello sviluppo di un sistema di processamento dei rifiuti solidi installato a Zagreb, Craozia. Il modello concettuale è realizzato utilizzando diagrammi ciclici attivi mentre per il modello di simulazione è stato impiegato il pacchetto software V56. La simulazione è risultata rilevante a causa della complessità del sistema e della sua elevata dinamicità. La verifica e la validazione del modello sono state effettuate in parallelo con lo sviluppo. Poiché il sistema reale non esiste, la validazione realizzata è di tipo *indipendent verification and validation* basata non su analisi statistiche ma sulla *face validity* (consultazione di esperti). Il modello al computer è stato verificato eseguendo esperimenti con il modello simulato per confrontarlo con quello concettuale. I fattori di valutazione, analizzati con tre differenti alternative di un piano fattoriale completo, sono il numero di inceneritori e di gru, ciascuno dei quali a due livelli. La simulazione è di tipo terminating e sviluppata a partire dalle

stesse condizioni iniziali. Sulla base dell'analisi di significatività dei fattori sono state selezionate solo due configurazioni del sistema tra cui effettuare la scelta. Tale scelta è stata realizzata valutando gli output di simulazione delle due alternative, attraverso analisi statistiche in termini di intervalli di confidenza per le medie degli indicatori.

*L'impiego di modelli di simulazione per il miglioramento delle performance di sistemi produttivi*

Il processo di miglioramento delle performance è un'analisi metodica del sistema nel suo complesso in termini di interazioni e interdipendenze degli elementi del sistema. In quest'ambito, sono stati considerati tre diversi approcci relativi all'impiego dei modelli di simulazione come strumento di supporto decisionale per la valutazione e il miglioramento di processi produttivi esistenti. Il primo caso è stato analizzato poiché permette di valutare la rilevanza della simulazione per il miglioramento delle prestazioni dei sistemi. Gli altri due approcci, invece, come esempio di utilizzo della simulazione per la risoluzione di problemi specifici nell'ambito della riprogettazione.
Per quanto concerne, l'analisi delle performance di sistemi esistenti un esempio è presentato da Alan et al.[6] Al fine di relazionare i cambiamenti nei componenti con le performance del sistema in analisi è necessario effettuare la modellazione del sistema. Gli autori analizzano il miglioramento delle performance del processo attraverso l'impiego della simulazione. Infatti, la simulazione fornisce informazioni circa gli elementi critici del sistema, le diverse interazioni e le sue dinamiche caratteristiche. I modelli di simulazione possono, quindi, essere impiegati per lo studio del comportamento del sistema attuale e la sua descrizione, per l'analisi degli elementi critici e per la stima delle performance e per valutare le diverse soluzioni proposte.

Un ulteriore esempio di valutazione delle prestazioni del sistema è quello presentato da Ueno et al. [150]. Come nel caso analizzato da Alan et al., si considera la simulazione come strumento di supporto alla riprogettazione del processo produttivo. Tuttavia, a differenza dell'approccio precedente, gli autori impiegano la simulazione non per effettuare una valutazione complessiva del sistema, ma per individuare i colli di bottiglia delle linee di produzione. In questo caso, si considera la simulazione come approccio alternativo a quello tradizionale di rilevazione della macchina con il tasso di produzione minore. Si dimostra che la tecnica impiegata risulta più realistica e pratica, in particolare nel caso in cui il sistema da analizzare sia ampio e complesso.

L'obiettivo della simulazione è la determinazione di una nuova configurazione della linea di produzione che permette di ottenere il livello di produzione desiderato con il minimo costo. In altri termini, si valuta la capacità produttiva, si identificano i colli di bottiglia per definire la nuova struttura della linea di produzione. Il modello è strutturato considerando un numero ridotto di sottomodelli poi connessi tra loro.

Gli input necessari per la realizzazione del modello sono:

- Caratteristiche macchine;
- sequenza job sulla prima macchina;
- Stoccaggio fuori linea e capacità buffer intermedi;
- probablità rilavorazione ad ogni ispezione;
- ispezione delle macchine e schedulazione delle riparazioni.

Il modello è generato utilizzando il linguaggio Slam e in misura limitata il Fortran.

La modellazione del sistema è particolarmente dettagliata, infatti, si considera anche l'impiego di macchine parallele, tenendo conto che il setup può essere effettuato solo per una macchina del gruppo per volta e le relative regole di assegnazione.

Le misure di performance sono i tassi di produzione attuali e non nominali(prodotti per ora) che tengono conto anche dei tempi di setup, di

rilavorazione di blocking e di starving. La procedura di simulazione è indicata in Figura 2.3.

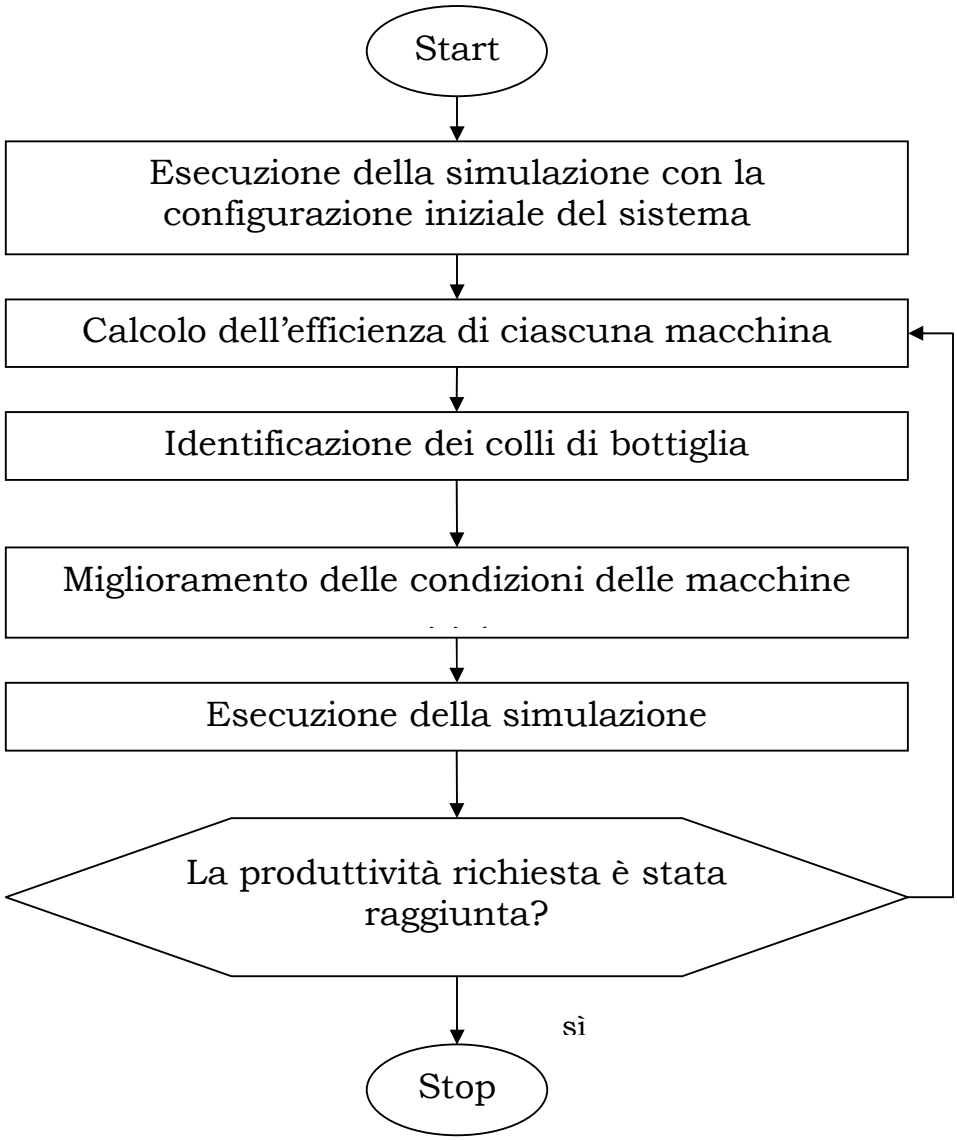



**Figura 2.3 Le fasi della simulazione**

Le condizioni sulle macchine su cui si interviene sono espresse in termini di tempo di processamento, di setup e tasso di rilavorazione. Quindi, l'obiettivo raggiunto dallo studio è stato la dimostrazione che la simulazione permette di individuare senza la necessità di impiegare elevate semplificazioni del sistema, le interconnessioni e le dipendenze tra le diverse parti del sistema.

Un ulteriore esempio di impiego della simulazione nell'ambito del processo decisionale strategico è stato presentato Kumar et al. [PIAN10] finalizzato alla riprogettazione del sistema di produzione al fine di migliorare la capacità produttiva. In particolare, gli autori hanno sviluppato di un sistema di simulazione DES per progettare una linea semi-automatizzata di produzione nell'ambito del processo di realizzazione di parti in plastica. In questo caso, la simulazione è impiegata al fine di migliorare il processo produttivo esistente in termini di capacità. Infatti, il modello di simulazione è impiegato prima per individuare i punti di debolezza del sistema, le condizioni di breakdown e i loro effetti sulla produzione e le capacità dell'impianto e per la valutazione dell'efficacia delle strategie di controllo. A questo punto, si ottimizza la produttività considerando diversi interventi migliorativi sul sistema. Il ruolo della simulazione è quello di valutare diversi interventi strategici alternativi e di incrementare le performance del sistema. In questa fase di valutazione delle possibili alternative, la simulazione è supportata dal Design of Experiment (DOE), che può essere impiegato per interpretare i risultati ottenuti e fornire informazioni su come specifici attori influenzano il comportamento del sistema. La validità dell'approccio proposto è stata dimostrata analizzando un processo con quattro fasi e buffer dedicati. La simulazione può essere, quindi, impiegata per valutare il comportamento del sistema e valutare i diversi possibili progetti alternativi. In realtà, gli autori implementano una sola tra le diverse alternative proposte, ovvero l'automatizzazione della linea di produzione con volumi maggiori. L'obiettivo principale dello studio di simulazione è stato quello di realizzare una "decisione informata" relativa alla capacità necessaria dei buffer in relazione all'automazione della linea di produzione. Inoltre, la simulazione ha permesso di analizzare gli effetti sul throughput conseguenti alla nuova configurazione del processo produttivo e di valutare le condizioni, in termini di definizione dei parametri, che permettono la massimizzazione del throughput per il nuovo assetto produttivo. Diverse tecniche come le superfici di risposta sono state impiegate per considerare il comportamento del sistema in termini degli obiettivi fissati al variare di alcuni fattori, quali il

tempo medio di ciclo delle macchine, la capacità del buffer, il mean time to failure delle macchine, il numero di macchine in parallelo per un dato stadio e la dimensione dei lotti.

### *2.8.4. L'impiego dei modelli di simulazione a livello tattico*

Un esempio di simulazione come strumento di supporto decisionale a livello tattico è mostrato da Watson et al.[175].Gli autori analizzano la pianificazione del rilascio degli ordini per una produzione make-to-order. In genere, tale pianificazione in un multi-stage shop è effettuata attraverso l'impiego della logica MRP, che assume capacità infinita delle risorse e lead time dei componenti stimati sulla base di dati storici e di esperienze passate. Queste assunzioni portano ad una pianificazione che spesso risulta irrealizzabile, il che implica elevate difficoltà nella realizzazione della fase di scheduling. Infatti, il MRP non considera in alcun modo che i lead time possono essere influenzati da diversi parametri come il carico di lavoro, la capacità le priorità degli ordini, il routing, la distinta base dei prodotti finiti, la dimensione del lotto e i vincoli del modello. Gli autori, al fine di sopperire a tali limitazioni, hanno proposto approccio alternativo, definito come *Resource Planning Based on Queuing Simulation* (qRP). Questo metodo genera il piano di rilascio degli ordini attraverso una logica di esplosione della distinta base backward simile a quella del MRP tranne che per l'impiego di un modello di un sistema di code simulato. Il modello simulato cattura il livello di dettaglio necessario per ottenere una rappresentazione più realistica. In tal modo, i lead time dei componenti sono time-based, ovvero dipendenti dallo stato attuale dello shop, e possono variare da periodo a periodo a differenza del MRP. Per realizzare questo tipo di pianificazione, è necessaria un'esplosione backward della distinta base dalle due date dei prodotti finiti al fine di stabilire le date adeguate di rilascio degli ordini. Questo processo può essere definito come *Simulation based order release planning* o *simulation based backward planning* o *simulation based resource*

*planning*. Il qRP si sostituisce, quindi, all'impiego del MRP/CRP nell'approccio tradizionale come mostrato in Figura 2.4.

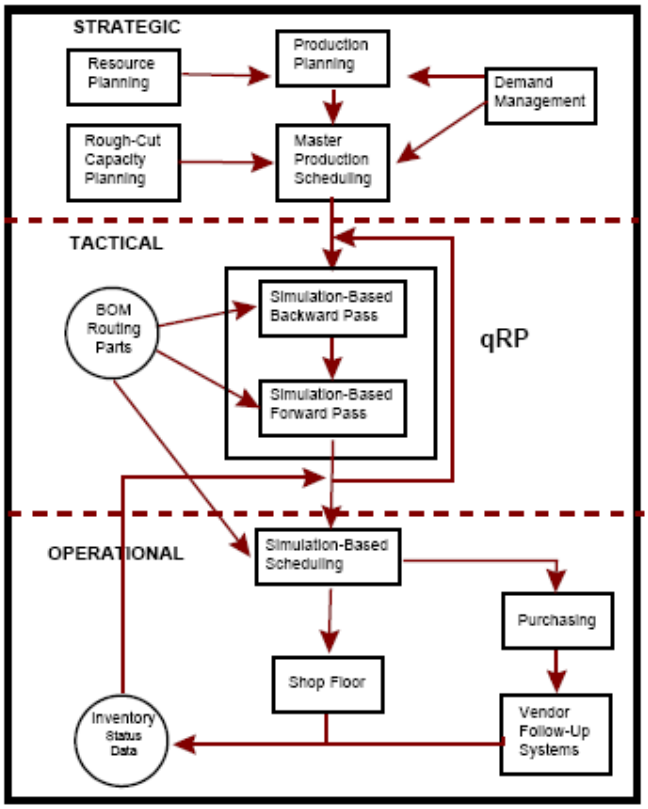



**Figura 2.4 -l'integrazione del qRP con il processo decisionale**

Tuttavia, al momento gli studi relativi a tale approccio sono molto ridotti e non esistono software commerciali che lo includano. Il qRP richiede un master schedule della domanda per i prodotti finali e informazioni sul processo quali il routing delle parti, le distinte base e i WIP, come il MRP/CRP e in aggiunta un modello di simulazione. L'approccio consiste in due fasi:

1. **Pianificazione backward** che determina i fabbisogni per gli ordini pianificati (effettivi e previsti) ;
2. **Pianificazione forward** che incorpora nella pianificazione gli ordini aperti.

La prima fase inizia ad un istante di tempo pari alla due date di ogni prodotto finito e li processa attraverso la sequenza inversa del routing delle parti. I tempi di rilascio di tutti i componenti primari così ottenuti costituiscono, a questo punto, gli input della seconda fase. Quest'ultima integra gli ordini aperti con quelli già pianificati nella fase precedente per fornire un piano fattibile. Le date di rilascio pianificate in precedenza possono essere considerate in questa fase come priorità. Oltre all'utilizzazione delle risorse, ci può non essere necessità di raccogliere statistiche o generare reports durante le due fasi. Le liste di dispatching sono create da un modulo di scheduling, ovvero lo schedulatore basato sulla simulazione. I modelli di simulazioni sono implementati con un linguaggio SIMAN V. Per tener conto dell'aleatorietà in termini di tempi di processamento, riparazioni e rottura delle macchine e handling di materiali è stato aggiunto un ulteriore modello, che rappresenta l'implementazione nella realtà, successivo ai due deterministici corrispondenti alle due fasi strutturato come quello forward. Per effettuare un confronto più realistico con il MRP, quest'ultimo è stato realizzato sulla base dei lead time forniti da un modello di simulazione forward e anche in questo caso gli output sono inseriti in un modello che considera l'aleatorietà. Il confronto è effettuato considerando la differenza delle misure di performance dei due sistemi. Al fine di fornire maggiore generalità, l'analisi comparativa è stata realizzata considerando diversi sistemi ottenuti da un generatore che definisce in maniera casuale l'ambiente di lavorazione e dal SIMAN V *model generator* che individua il sistema di produzione in termini di layout, distinta base, prodotti finiti e routing dei componenti. In particolare, è stata effettuata la sperimentazione con un piano fattoriale completo caratterizzato dai seguenti fattori: struttura del prodotto(piatta, elevata, complessa), shop flow (flow shop, job shop), modello di domanda del master scheduler ( stabile, instabile), shop load (leggero pesante) variabilità del sistema (bassa, alta) bilanciamento ( un collo di bottiglia, più colli) assegnazione della due date (costante, basata su regole di lavoro) e carico di lavoro iniziale (leggero, pesante).

Le misure di performance sono il ritardo medio degli ordini, la percentuale di ordini in ritardo, anticipo medio, numero medio di WIP, lead time medio Dalla sperimentazione si è dimostrato che il qRP è migliore se non ci sono grosse variabilità degli ordini, anche se tale vantaggio si riduce in maniera non consistente nel caso di un prodotto complesso e le regole non sono FIFO. In conclusione, i benefici del qRP diminuiscono se lo shop è volatile e anche nel caso in cui la regola di dispatching non è FIFO, in quanto questa è la regola impiegata nelle due fasi. Quindi, tale regola risulta un fattore chiave. Quindi, il qRP è particolarmente vantaggioso se la domanda può essere prevista in maniera accurata almeno per un periodo in anticipo, quando ci sono larghe variaizioni nell'utilizzazione delle risorse da periodo a periodo, quando il carico dello shop è relativamente alto, quando la struttura del prodotto ha molti livelli e componenti da assemblare. In questi ambienti il qRP è migliore del MRP.

L'approccio considerato è uno solo tra i possibili esempi applicativi relativi all'impiego della simulazione a livello tattico. In particolare, l'analisi effettuata da Watson et al. è stata selezionata in quanto dimostra come l'introduzione dei modelli di simulazione permette il miglioramento delle prestazioni in maniera del tutto indipendente dal tipo di processo produttivo implementato.

### *2.8.5. L'impiego dei modelli di simulazione a livello operativo*

Molti autori hanno analizzato l'impiego della simulazione come uno strumento di gestione sviluppato per supportare il processo di *operation decision making*. Questo permette di analizzare in termini statistici il comportamento del processo sotto definite condizioni, in genere costituite da fattori controllabili e non controllabili. La simulazione permette di selezionare le decisioni operative che permettono il raggiungimento degli obiettivi e di valutare gli effetti di queste decisioni in relazione alla modifica

nel tempo dei fattori non controllabili, ovvero dell'ambiente. Quindi, è possibile con un insieme di condizioni predefinite e ordini simulare il processo e anticipare il risultato di tale sistema operativo. Laddove si verifichino delle problematiche, quali colli di bottiglia o sovraccarichi, è possibile individuare soluzioni alternative che migliorino le performance del sistema.[110] Per quanto concerne l'impiego della simulazione come strumento di supporto decisionale operativo, è possibile considerare diverse problematiche analizzate. In particolare, sono stati considerati approcci mirati alla valutazione di problemi di schedulazione del lavoro. Inoltre, si è considerato l'impiego dei sistemi di simulazione nell'ambito della schedulazione dei processi produttivi. Per quanto concerne questo tipo di analisi sono stati differenziati gli approcci mirati alla schedulazione dei sistemi di tipo flow shop da quelli di tipo job shop. Un'ulteriore analisi proposta nell'ambito operativo è quella relativa all'mpiego della simulazione come supporto al processo decisionale. Infine, sono stati valutati i sistemi integrati proposti in letteratura, come esempio di sviluppo di strutture più o meno centralizzate focalizzate sull'impiego della simulazione.

*L'impiego dei modelli di simulazione per la schedulazione giornaliera del lavoro.*

Un ulteriore modalità di impiego della simulazione ad eventi discreti in ambito operativo è quello fornito da Andersson et al. [13]. In questo caso, gli autori impiegano la simulazione per pianificare l'allocazione giornaliera delle risorse umane. L'articolo considera il sistema di supporto decisionale basato sulla simulazione (SBDS) a livello operativo del sistema di produzione. Il caso analizzato è quello di una linea di assemblaggio Radio Base Station (RBS) alla Ericsson Radio System, Gävle. Lo scopo del lavoro è quello di determinare come la DES può essere impiegata nell'ambito delle decisioni operative per l'assegnazione del lavoro giornaliero in una linea di assemblaggio customer-driven, considerando tale sistema come isolato dagli

altri sistemi. Il modello di simulazione è usato per prevedere la precisione di spedizione e l'impiego delle risorse in maniera giornaliera, supponendo che l'utilizzatore del SBDS sia il supervisore della forza lavoro.
Il modello di simulazione è stato sviluppato con il pacchetto Taylor II per Windows. Le variabili di performance considerate sono la precisione di spedizione di ordini di produzione separati, l'utilizzazione delle risorse umane e delle stazioni operative. L'articolo analizza le diverse fasi dello sviluppo del sistema di simulazione. Tuttavia, ai fini della valutazione della simulazione come strumento di supporto al processo decisionale, si è analizzato con maggior dettaglio lo sviluppo del modello di simulazione. Per quanto riguarda la validazione del modello di simulazione, questa è effettuata attraverso l'impiego di animazioni, in quanto l'obiettivo dello studio era quello di valutare l'impiego della simulazione come sistema di supporto decisionale e non l'ottenimento di un sistema operativo. Il sistema è interfacciato con altri sistemi, ovvero con il decisore, con il Manufacturing, Planning and Control System (MPCS) e con la linea reale di assemblaggio. La simulazione è stata testata per 8 settimane considerando le due linee di assemblaggio identiche al fine di simularne una sola. Sono state analizzate cinque replicazioni per ogni giorno. Inoltre, se dalla simulazione la precisione di spedizione risultava la di sotto del 100% sono state considerate le possibili azioni di intervento in termini di rischedulazione del lavoro. I risultati ottenuti evidenziano la capacità di tale approccio di migliorare le performance del sistema senza modificare il quantitativo totale di risorse impiegate. La Figura 2.5 rappresenta il flusso informativo del processo decisionale ottenuta.

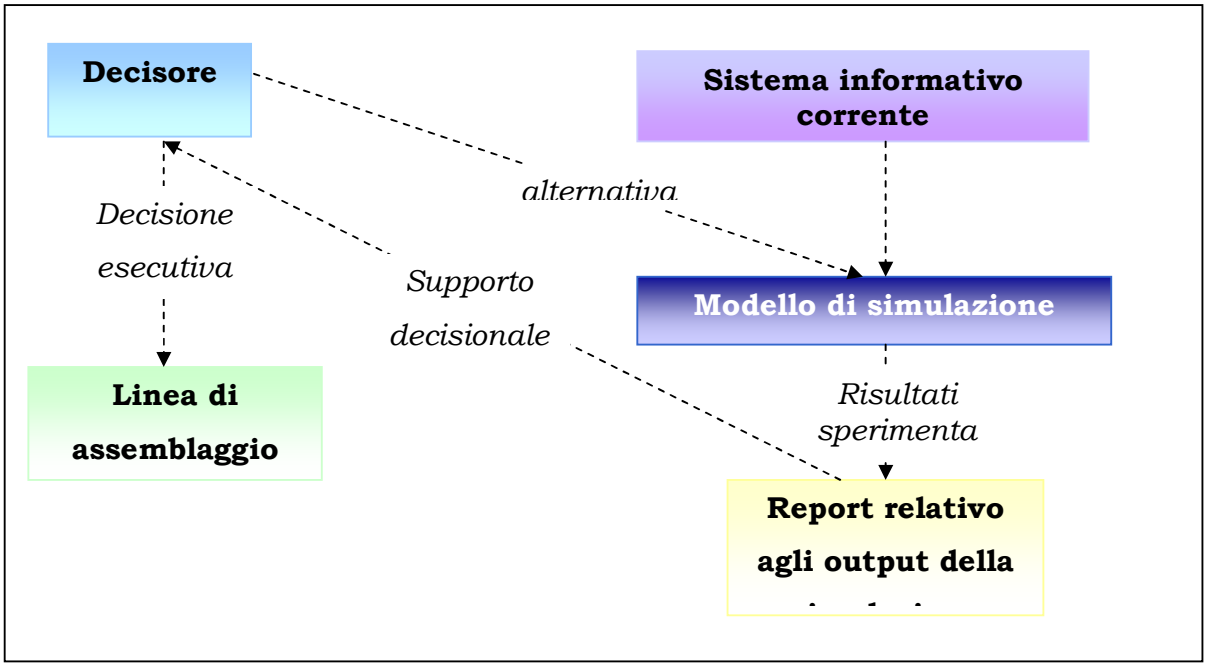



**Figura 2.5 Flusso informativo**

Il decisore ha una serie di alternative che sono realizzate dal sistema di simulazione. Dai report sulla simulazione è possibile selezionare l'alternativa ritenuta migliore e implementarla nel sistema reale. La limitazione di tale approccio è la necessità di avere dati accessibili in tempo reale sullo stato effettivo giornaliero della linea di assemblaggio. Ad oggi tali informazioni non possono essere ottenute in maniera automatica. Quindi, anche se il modello di simulazione si interfaccia direttamente con il sistema informativo aziendale, acquisendo, tuttavia, da questo solo parte delle informazioni necessarie, non è possibile considerarlo come un sistema integrato perché risulta non completamente connesso con il processo reale.

*L'impiego dei modelli di simulazione per la schedulazione dei processi produttivi.*

Particolare attenzione, nell'ambito della simulazione ad eventi discreti come sistema di supporto decisionale operativo, è posta per i problemi di scheduling della produzione

Come modalità di modellazione utilizzata per la schedulazione, la DES ha tre principali caratteristiche. Prima di tutto permette di ottenere schedulazioni stabili attraverso l'approccio incrementale, ovvero, a differenze delle tradizionali tecniche di scheduling, piccoli cambiamenti nei parametri del sistema non variano sostanzialmente la schedulazione effettuata. Inoltre, permette la trattazione di problemi complessi senza particolari semplificazioni. Il livello di accuratezza dei modelli è fondamentale soprattutto nell'ambito della lean production, in cui l'effetto delle interazioni tra componenti del sistema è crescente al ridursi del lead-time e dei WIP. Un modello di simulazione, infine, può essere impiegato anche come strumento di analisi preliminare della struttura del sistema oltre che di pianificazione operativa.

Nell'ambito della schedulazione, è possibile individuare diversi studi effettuati sia per processi flow shop che job shop. Poiché il primo può essere analizzato come un caso particolare del secondo, maggiore attenzione è stata rivolta all'impiego di modelli di simulazione per il job shop.

*L'impiego dei modelli di simulazione per la schedulazione di sistemi flow shop*

Un esempio di schedulazione per il flow shop è quello proposto da Vaydianathan et al. [169]. Tra i numerosi approcci presenti in letteratura, questa analisi è stata valutata in quanto particolarmente accurata. Gli autori descrivono l'impiego della DES nell'industria di processo per la schedulazione giornaliera. Un largo numero di prodotti finali (circa 300), domanda sporadica e shelf-life limitata (90gg) rendono difficile la generazione manuale della schedulazione. Il sistema sviluppato è caratterizzato da due parti, ovvero dal programma di schedulazione e dal modello di simulazione. Il primo è impiegato per generare la schedulazione giornaliera. Il sistema di

simulazione, invece, usa la schedulazione ottenuta per simulare il sistema e migliorarla. Il caso applicativo considerato è caratterizzato da un processo di produzione di caffè costituito da quattro principali fasi con capacità limitata. Un'ulteriore fase è lo stoccaggio per il "degas", ovvero il rilascio dell'anidride carbonica contenuta dal caffè e richiede almeno ventiquattro ore. Inoltre, gli autori hanno considerato vincoli di produzione dovuti ad una shelf-life limitata del prodotto e una domanda fortemente stagionale, con picchi nei quattro mesi invernali nel caso applicativo. Gli autori evidenziano la rilevanza della simulazione nell'ambito dello scheduling, considerando la complessità del sistema in termini di:

- Lead time di produzione molto lunghi, in particolare per il processo esaminato variano dalle ventiquattro alle trentasei ore senza nessun altro prodotto in competizione per le scarse risorse;
- Capacità limitata e, nel caso specifico, punti di stoccaggio intermedi per il degassing sono limitati;
- Stagionalità della domanda;
- Prodotti omogenei, quasi tutti i produttori fanno lo stesso prodotto, quindi, la soddisfazione del cliente è critica;
- Le risorse critiche, ovvero con maggiori vincoli di capacità, variabili a seconda della schedulazione. Quindi, non possono essere isolate e non si può effettuare la massimizzazione della loro utilizzazione per massimizzare la produzione complessiva;
- Necessità di rischedulare per considerare breakdown, arrivi di larghi ordini nuovi.
- Impossibilità di incremento della capacità perché questa è adeguata alla domanda nei periodi non di picco.

La prima schedulazione è realizzata senza considerare le interazioni tra prodotti e risorse. Il modello simula il processo e effettua modifiche in termini di quantità e sequenza sulla base dello scenario giornaliero. Una traccia completa della simulazione è, in tal modo, ottenuta e la sequenza in cui il modello effettua la schedulazione è assunta come nuova

schedulazione. Inoltre, la valutazione statistica degli output permette all'analista di effettuare ulteriori modifiche.

Il sistema è stato sviluppato con Visual Basic. I dati in input principali sono gli ordini, le previsioni la scorta di prodotti finiti e la scorta di WIP. La schedulazione è effettuata secondo un pull system, ovvero schedulando per prima la fase finale. La dimensione dei job è determinata sulla base della domanda. Ovvero, i job con minore domanda hanno dimensione pari alla quantità ordinata. Per quelli, invece, con domanda elevata si produce una quantità pari agli ordini noti più le previsioni entrambe per i successivi tre giorni. Sulla base di tale schedulazione si procede a ritroso a schedulare le fasi precedenti del processo. La schedulazione per ogni fase e il livello di scorte dei WIP sono gli input del modello di simulazione sviluppato con SIMAN. Il modello elabora le informazioni e genera i reports statistici e il file di schedulazione modificato. Il decisore può, sulla base di tali informazioni, modificare la schedulazione finale e ripetere la simulazione fino a che lo scheduling non è approvato e implementato. Il modello di simulazione è costituito da quattro moduli corrispondenti alle quattro fasi del processo produttivo. La prima fase genera una schedulazione sulla base dei dati disponibili per evitare situazioni di starving nella fase successiva. A questo punto, le fasi successive realizzano la schedulazione generata dallo schedulatore in funzione dei parametri attuali del modello. In realtà, sono stati realizzati due modelli in quanto si effettua una diversa schedulazione per la settimana e il week-end. I due sistemi sono identici, tuttavia variano alcuni parametri operativi come le ore macchina. Il modello di simulazione è stato validato attraverso il confronto degli output su un periodo di due settimane con gli output del sistema attuale nello stesso periodo. La verifica è stata effettuata in maniera manuale. Il processo di scheduling proposto è mostrato in Figura 2.6.

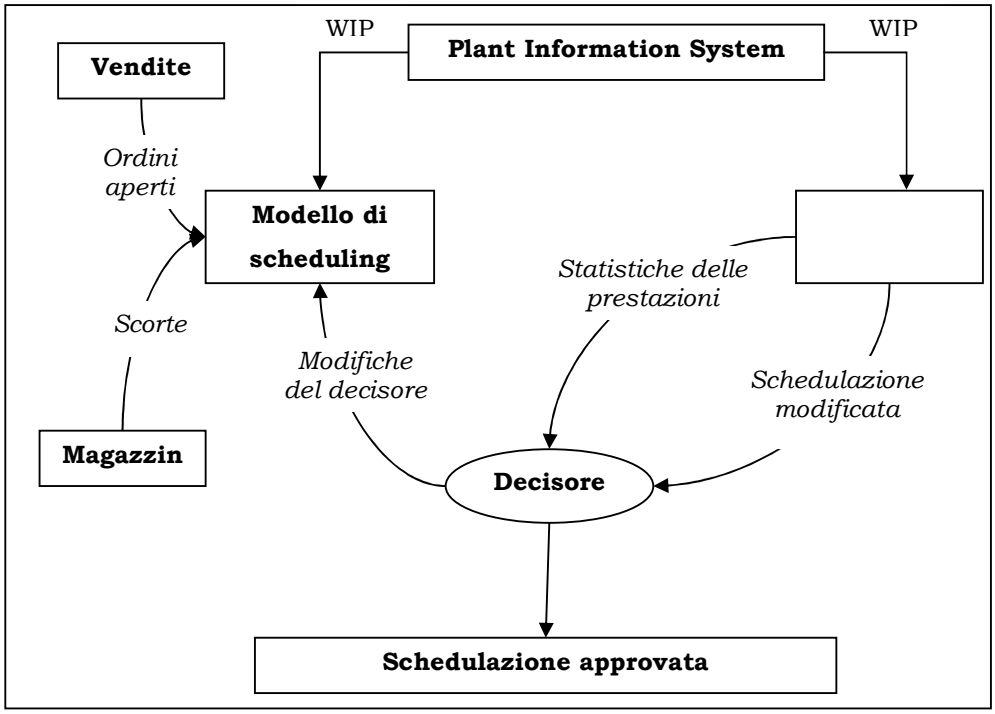



**Figura 2.6 Il processo di scheduling**

Dall'analisi dei risultati sperimentali si evince che l'impiego della simulazione per realizzare la schedulazione del processo permette di migliorare gli indicatori di performance valutati, in termini di tassi di utilizzazione e quantità prodotte.

*L'impiego dei modelli di simulazione per la schedulazione dei processi di tipo job shop*

Per quanto concerne l'analisi nell'ambito del job shop, molti approcci possono essere considerati. Le prime due tecniche di schedulazione valutate, proposte da Backer et al.[17] e Palaniswami et al.[131] mostrano l'efficacia dell'impiego della simulazione come semplice strumento di supporto decisionale. In particolare, il primo studio impiega un numero elevato di ipotesi semplificative nella valutazione del sistema. In altri termini, l'obiettivo dell'analisi è la dimostrazione dei vantaggi derivanti dall'impiego della simulazione, che permette un'analisi più realistica in termini di aleatorietà del processo. Tuttavia, il modello simulato risulta comunque molto

semplificato. Quindi, in relazione anche all'anno di pubblicazione (1960) tale studio è stato proposto come possibile punto di partenza per valutare l'evoluzione della tecnica. Per questo motivo, è stato successivamente considerato l'analisi realizzata da Palaniswami et al. [131], in quanto, in questo caso, il numero di ipotesi semplificative impiegate è molto ridotto. In entrambi i casi, comunque, la simulazione è utilizzata come strumento di verifica della schedulazione effettuata. Diverso, invece, è l'approccio proposto da Selladurai et al.[151], che realizzano la schedulazione direttamente impiegando la simulazione. Tuttavia, tale studio è limitato all'impiego di metodi di schedulazione basati su semplici regole di dispatching. Negli ultimi anni, invece, i sistemi di simulazione per la schedulazione nell'ambito job shop sono diventati sempre più complessi al fine di soddisfare specifiche esigenze produttive. In particolare, sono state analizzate due problematiche. La prima consiste nella necessità di sviluppare sistemi di simulazione che permettano la schedulazione multi-obiettivo con tempi di setup variabili. In questo caso, la simulazione risulta un mezzo di analisi fondamentale in quanto permette di scomporre problemi NP-hard in sottoproblemi. Alcuni esempi di analisi di questo tipo di problematica sono stati proposti da Yang et al. [181], che valutano l'impatto di un approccio di ottimizzazione multi-obiettivo basato su un'analisi di Pareto attraverso la simulazione nell'ambito di sistemi produttivi come quelli di realizzazione di circuiti integrati. In particolare, sono considerati diversi esempi numerici confrontando la metodologia proposta con l'approccio tradizionale con un obiettivo e diverse regole euristiche di dispatching. Altri esempi sono quelli proposti da Sivakumar [155] e Gupta e Sivakumar [72] con particolare attenzione alla produzione di semiconduttori. In particolare, sarà analizzato nel dettaglio uno studio più esaustivo, in quanto analizza casi sperimentali reali e confronti con le altre tecniche di schedulazione come i metodi euristici, presentato da Gupta et al. [71].

Una seconda problematica affrontata negli ultimi anni è quella relativa alla realizzazione di una schedulazione che permetta di minimizzare il ritardo degli ordini impiegando un sistema di simulazione di tipo

Backward/Forward Hybrid Simulation (BFHS), al fine di ottenere un livello di dettaglio e di accuratezza elevato. In particolare, primi esempi di impiego di questo tipo di simulazione sono stati proposti da Fuyuki et al. [PIAN31]. Gli autori hanno dimostrato che, su cento problemi di job shop semplici per diversi livelli di ristrettezza della due-date, il metodo proposto risulta in gran parte dei casi migliore rispetto alle tecniche tradizionali. Tuttavia, gli autori hanno valutato il BFHS per minimizzare il ritardo e non gli scostamenti globali rispetto alla due-date, in termini anche di anticipo oltre che di ritardo. Tale tecnica è stata per questo scopo perfezionata da Arakawa et al. [16], utilizzando un'ulteriore tipologia del BFHS definito type D che combina il type C e quello B. Infatti, Fuyuki et al. [PIAN31] hanno impiegato il type C, in cui l'ordine di tempo di inizio di lavorazione per ogni job su ogni macchina ottenuto dalla simulazione backward è utilizzato come ordine di priorità dei job nella simulazione forward. Il type B, invece, usa i tempi ottenuti dalla simulazione backward per controllare l'assegnazione dei job nella simulazione forward, in modo da migliorare le prestazioni della tecnica adoprata. In relazione alla migliori prestazioni e al fatto che la minimizzazione della deviazioni rispetto alla due-date permettono di soddisfare le esigenze in termini di scorte finali molto contenute e pochi WIP, soprattutto nell'ambito della produzione make-to-order per sistemi di produzione snella, ad oggi largamente diffusi, si è deciso di analizzare l'approccio proposto da Arakawa et al. [16] con maggior dettaglio.

Il primo studio valutato è quello proposto da Baker et al.[17].Il modello simulato dagli autori rappresenta un job shop con numero ridotto di risorse (da 9 a 30) ciascuna diversa dalle altre e capace di processare un solo job per volta. Il tempo di processamento è considerato aleatorio per tener conto di eventuali scostamenti dalla schedulazione effettuata.

I dati in input sono le informazioni sul routing e i tempi di processamento generati con generatori di numeri pseudo-casuali. I parametri controllati sono la dimensione del job in termini di numero di risorse e il numero medio di operazioni di processamento per ogni job.

Si considerano due tipi di carico:

a) tipo A: le operazioni sono organizzate in modo che nessuna macchina sia vuota e nessun job in attesa.
b) tipo B: schedulazione imperfetta.

I fattori considerati sono il tempo effettivo di processamento e tempo atteso, il primo calcolato in funzione del secondo. Il modello è sviluppato come un sistema di code con la dimensione del job fissata pari a *m*. L'analogia con un sistema di code è ottenuta considerando il livello di produzione, ovvero la percentuale rispetto al massimo, come il "fattore di utlizzazione" per il sistema di code e il tempo medio totale di lavorazione del job pari alla somma dei tempi medi di attesa in coda e quelli medi di servizio.

Le ipotesi semplificative considerate sono:

- Un carico di lavoro atteso uguale su tutte le risorse per evitare colli di bottiglia complessi;
- Due operazioni successive non possono essere realizzate sulla stessa macchina;
- Il numero di parti uguali lavorate come un unico job non sono considerate;
- I tempi setup o transitorio non sono valutati come variabili separate;
- Non è previsto l'order splitting;
- Il sistema è supposto a capacità fissata, ovvero non considerano straordinari o turnover del personale;
- Non si considerano macchine parallele, ovvero non ci sono gruppi di risorse uguali;
- Non sono previste politiche di accumulo jobs con setup simili su una risorsa per ottimizzare i tempi di setup. In altri termini, i job non sono suddivisi in famiglie.

Al fine di analizzare lo stato stazionario,si considerano le performance dopo che venti job sono stati rilasciati dal sistema. Le misure di performance considerate sono i tempi totali di produzione e tempi di completamento e il rapporto tra tempo di lavorazione effettivo e atteso per ogni job come indice di predicibilità.

Per ogni tipo A e B si considera la sperimentazione di un piano fattoriale $3^3$ in funzione di tre fattori:

1) Dimensione dello shop;
2) Tasso di variazione del lavoro;
3) Numero medio di operazioni di processamento per job.

Per ogni tipo di carico si valutano politiche di schedulazione basate su regole di dispatching FIFO e Random. Si analizza, inoltre, il comportamento del sistema considerando sia regole di risoluzione dei conflitti semplici e poi metodi più complessi.

Il modello considerato risulta, quindi, piuttosto approssimativo e poco rappresentativo della realtà. Tuttavia, tale approccio evidenzia una prima caratteristica fondamentale conseguente all'impiego della simulazione come supporto decisionale alla realizzazione della schedulazione, ovvero la possibilità di valutare a differenza dei metodi analitici, anche se solo in termini di fluttuazione dei tempi di processamento, l'aleatorietà intrinseca dei processi di produzione.

Palaniswami et al. [131] hanno analizzato i modelli di simulazione per testare l'efficacia di una o più regole di dispatching. Noto che le semplici regole di priorità considerano variabili quali la due date e i tempi di processamento per i job; la crescente complessità dei sistemi e le tecnologie innovative richiedono la valutazione di ulteriori variabili nel modello di simulazione per effettuare un'analisi più realistica. In primo luogo, è necessario analizzare la complessità di alcuni job che causa maggiori tempi di setup e di produzione. Inoltre, altre variabili quale la diminsione dei lotti, i breakdown delle macchine, la carenza di scorte e gli errori degli operatori possono influire sulle prestazioni del sistema. Tutte queste variabili possono essere organizzate in maniera gerarchica. Il primo insieme di variabili, ovvero dimensione dei lotti, tempi di setup e processamento, che sono uniche per ciascun job, possono essere considerate ad un primo livello. Il secondo insieme di variabili, ovvero il breakdown delle macchine, la carenza di materiali e gli errori possono essere considerate ad un livello inferiore e sono influenzate dall'efficienza dell'ambiente di produzione e si verificano in

maniera casuale durante la produzione. La simulazione discreta permette di rappresentare un modello di simulazione per un sistema job shop considerando tali variabili. In ogni caso, la validità e l'utilità di tale modello dipende dalla stima realistica di tali variabili, che spesso può essere realizzata da esperti del sistema reale o, in maniera più efficiente, attraverso l'impiego di un sistema esperto. Il decisore può, quindi, determinare lo scheduling più adeguato sulla base dei risultati della simulazione i cui input sono forniti dal sistema esperto come mostrato in Figura 2.7.

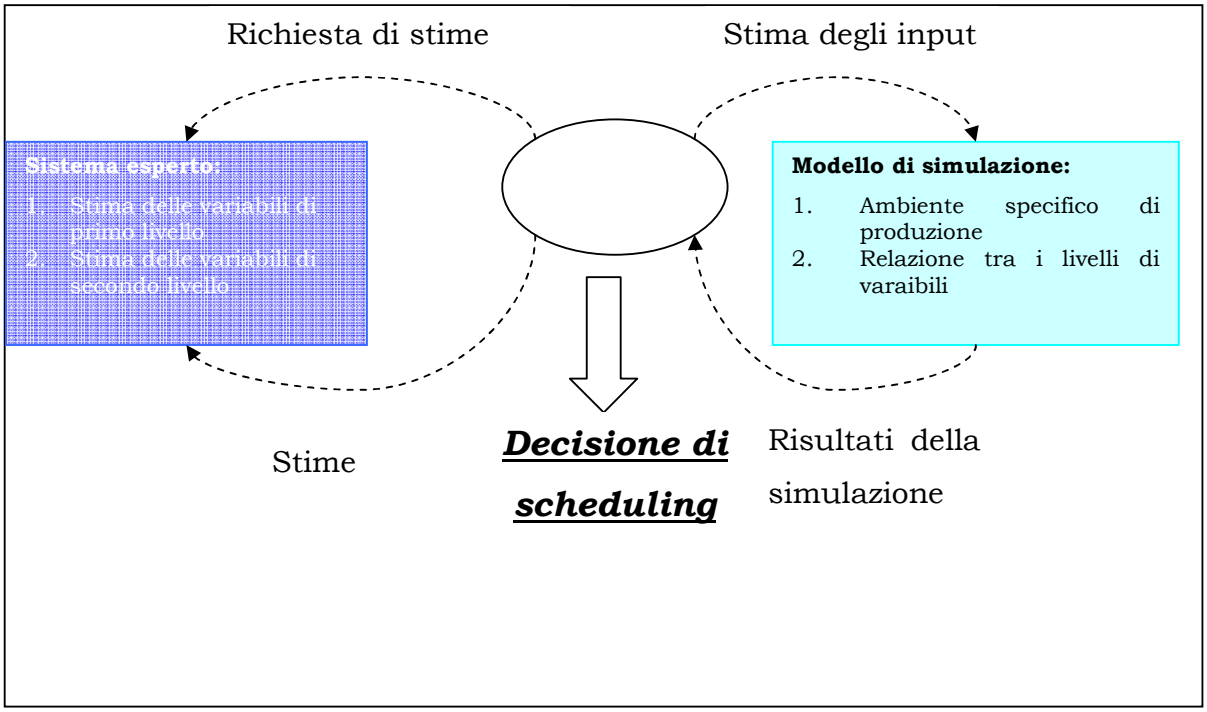



**Figura 2.7 Processo decisionale per la realizzazione della schedulazione**

Il sistema esperto si basa sulla conoscenza dell'esperienza passata in relazione al sistema e alle tipologie di job completate attraverso la quale realizza una stima realistica delle variabili.

Il modello di simulazione rappresenta un particolare ambiente di produzione in termini di flusso di processamento e relazioni dinamiche tra i due insiemi di variabili per un particolare insieme di job. In tal modo, il decisore può

valutare lo scheduling più realistico per un dato intervallo di tempo T. Il sistema di simulazione è mostrato in Figura 2.8.

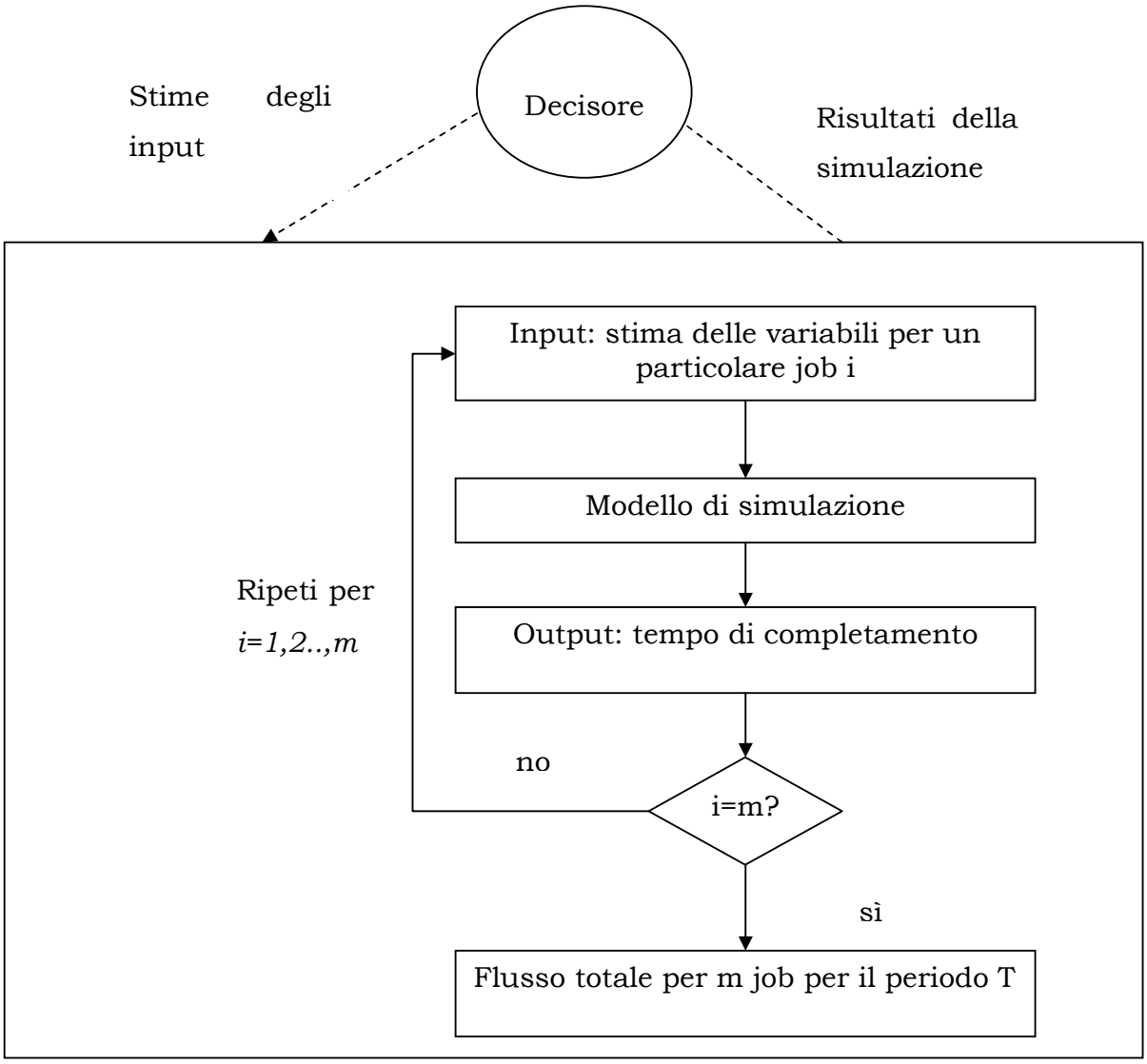



**Figura 2.8 Sistema di simulazione impiegato**

Il sistema esperto utilizzato è il pacchetto M.1 . Il modello di simulazione può essere sviluppato impiegando i linguaggi di programmazione convenzionali o altri linguaggi di simulazione come lo SLAM II o il SIMSCRIPT. Il sistema è stato implementato per un processo ipotetico con due stazioni in serie che

lavorano quattro job, i quali sono caratterizzati da tre livelli di complessità che incidono sulla stima delle variabili di primo livello. Il modello di simulazione è stato realizzato utilizzando SLAM II che permette di ottenere statistiche in output quali la lunghezza delle code, i tempi di attesa e i tempi di completamento comprendere se rientra nel tempo previsto. Inoltre, nella simulazione è stato considerato il breakdown di una delle stazioni che si verifica con distribuzione esponenziale degli interarrivi. Anche i tempi di riparazione sono esponenziali. In breve, l'applicazione dimostra l'uso di un sistema esperto come interfaccia tra il modello di simulazione e il decisore . In questo caso, quindi, la simulazione è impiegata per valutare il processo produttivo e le relazioni dinamiche tra i due livelli di variabili per i job da schedulare. In particolare, il modello di simulazione permette di effettuare una valutazione più realistica del sistema reale rispetto allo studio precedente, in quanto considera, non solo l'aleatorietà del sistema in termini di variazione dei tempi di processamento, ma anche valutando la presenza di eventi completamente casuali quali il breakdown delle macchine. Inoltre, le iotesi semplificative adottate sono molto più verosimili rispetto a quelle proposte da Baker et al. [17].

Un approccio alternativo è presentato da Selladurai et al. [151]. Lo studio considerato risulta particolarmente interessante per due motivi. In primo luogo, come nel caso precedente, sono considerati eventi aleatori difficilmente analizzabili con le tecniche di scheduling puramente analitiche. Tuttavia, in questo caso gli eventi casuali non sono i possibili breakdown delle macchine, che sono trascurati, ma l'urgenza o meno degli ordini. Il secondo fattore da considerare, invece, è l'evoluzione rispetto al caso precedente in quanto in questo studio è prevista la possibilità di effettuare direttamente in maniera automatica durante la simulazione la schedulazione. Bisogna, però, considerare che le tecniche di scheduling impiegate sono molto semplici e poco efficienti per sistemi complessi. In questo caso, si analizza la simulazione per studiare gli effetti delle regole di schedulazione e di priorità considerando o meno ordini urgenti valutando diverse possibili funzioni obiettivo alternative, come la minimizzazione delle

scorte di WIP o del tempo di attesa oppure la massimizzazione dell'utilizzazione delle risorse o del throughput. La schedulazione è effettuata assumendo che ciascun componente sia lavorato solo da un centro e che ciascun centro lavora un componente alla volta. Inoltre, i componenti non possono essere splittati e non sono considerati breakdown delle macchine. La simulazione è effettuata considerando sei centri di lavoro e a seconda della probabilità assegnata sono trattati i job come urgenti o normali. Questa distinzione influisce sulla selezione del centro di lavoro. I tempi di arrivo e di processamento hanno distribuzioni definite dall'utente sulla base di dati storici. L'analisi è effettuata con una schedulazione pre-emptive e non pre-emptive se sono presenti ordini urgenti, solo pre-emptive se ci sono solo ordini non urgenti. Il linguaggio di simulazione è block-oriented ed è il GPSS/PC. Il modello è analizzato nel caso in cui tutti gli ordini siano normali. Inoltre, si è analizzato il caso in cui ci siano anche ordini urgenti con priorità pre-emptive e non pre-emptive e considerando solo due regole di schedulazione SPT e FIFO assegnate automaticamente dal sistema o esternamente. Infine, il modello è stato valicato attraverso la valutazione di intervalli di confidenza. Lo studio ha mostrato che alcune regole, per esempio la HRN è migliore nel caso di ordini non urgenti rispetto alla FIFO e SPT in termini di utilizzazione e minore variabilità, mentre lo è la FIFO nel caso di ordini urgenti. Per i WIP la SPT è migliore nel caso di non-preemptive e ordini urgenti, per la schedulazione pre-emptive è migliore la FIFO. Infine, la SPT è migliore per la massimizzazione del throughput.

Concludendo, anche sulla base dell'analisi dei tre casi proposti è possibile affermare che l'evoluzione delle tecniche di simulazione e degli studi relativi ha permesso di superare le limitazioni, nell'ambito dello scheduling del processo, dovute al netto scostamento rispetto alla realtà che in genere si verifica quando si impiegano tecniche puramente analitiche.

Per quanto concerne la necessità di sviluppare sistemi di simulazione che permettano la schedulazione multi-obiettivo con tempi di setup variabili, è stato analizzato l'approccio proposto da Gupta et al. [71]. Gli autori presentano un'analisi preliminare della schedulazione per la produzione di

semiconduttori comprendendo più famiglie di job, nel caso esaminato sono due con $N$ job totali su una macchina per i test. In particolare, il problema analizzato è di tipo NP-Hard in relazione ai setup time dipendenti dalla sequenza e al problema multi-obiettivo. In altri termini, i tempi di setup variano al variare dei job appartenenti alla stessa famiglia e ulteriormente al variare della famiglia. L'unica ipotesi adottata è che non siano possibili job splitting per gli stringenti vincoli di qualità. Gli obiettivi sono la minimizzazione del tempo medio di ciclo e del ritardo e la massimizzazione dell'utilizzo delle risorse. Una soluzione ottima di Pareto non inferiore a nessun altra soluzione realizzabile per tutti gli obiettivi, è ottenuta combinando l'ottimizzazione analitica e lo scheduling basato sulla simulazione. Il problema di scheduling è modellato usando la DES dividendolo in due sottoproblemi di selezione di tipo simulation clock- based. Quindi, un lotto ottimo di Pareto è selezionato utilizzando la tecnica di programmazione basata sul compromesso, che considera la soluzione più vicina a quella ideale in termini di distanze relative per l'ottimizzazione multi-obiettivo ad ogni istante decisionale nel tempo simulato. La DES permette di non affrontare un problema NP-hard poiché il problema complessivo è scomposto in sottoproblemi consistenti nella selezione dei job per ogni centro locale e ad ogni istante decisionale. Ad ogni istante è selezionato il job in coda che presenta il minimo valore della funzione di compromesso. A questo punto il clock di simulazione è incrementato del tempo di setup e processamento. Il sistema è stato valutato con un piano fattoriale $2^5$ dove i fattori sono il numero di job, uguale per ogni famiglia, i tempi di setup da un job ad un altro, la matrice di quelli al variare della famiglia, i tempi di processamneto e le due date ciascuno a due livelli. La soluzione sviluppata, inoltre, è stata confrontata con le più comuni dispatching rules euristiche come la SPT e EDD. Il confronto ha determinato migliori performance in termini di ritardo medio, tempo medio di ciclo e uso delle risorse perché la presenza di più famiglie cambia sostanzialmente i tempi di setup e non è considerata dalle regole tradizionali. Sebbene l'analisi non sia stata ancora effettuata sulla base di dati reali, l'approccio

considerato risulta particolarmente efficace. Infatti, tale studio permette di superare le spesso eccessive semplificazione in merito ai tempi di setup e, soprattutto, di realizzare un'ottimizzazione multi-obiettivo senza dover ricorrere a rappresentazioni troppo approssimative del sistema.

Infine, si è considerato lo studio proposto da Arakawa et al. [16] basato su un metodo di minimizzazione della deviazione rispetto alla due date degli ordini sulla base della Backward/Forward Hybrid Simulation (BFHS) in un processo di tipo make-to-order. Un nuovo metodo di schedulazione definito come il BFHS/type-D è stato sviluppato. Tale tecnica si basa sulla realizzazione di una simulazione backward, le cui informazioni sono impiegate per definire le priorità dei job in quella forward.

Il metodo type D è caratterizzato da tre fasi:

1. ***Backward simulation*** realizzata sul modello di produzione tracciando i cambiamenti di stato backward, ovvero nella sequenza inversa sulla scala temporale. La procedura inizia in un punto del tempo futuro che è, in genere, la due-date più ampia tra i job da schedulare. Il job con la due-date più lontana è scelto e l'operazione finale di questo assegnata a una macchina nel centro di lavoro selezionata in modo che il tempo di completamento dell'operazione coincida con la due-date e si calcola il tempo di inizio dell'operazione. Le operazioni finali degli altri job sono assegnate sulle macchine disponibili in maniera sequenziale. Considerando i tempi di trasporto tra le stazioni, si calcola il tempo di fine delle operazioni precedenti sulla base dei tempi di inizio di quelle già schedulate. La procedura è realizzata fino all'assegnazione di tutte le prime operazioni. Se alcuni job sono in conflitto su una macchina si considera l'impiego delle regole di dispatching;
2. ***Estrazione delle informazioni richieste***, ovvero dai risultati della simulazione backward si estraggono i tempi di inizio di tutte le operazioni per ogni job e sono trasformate in due tipi di operazioni:
    - Condizioni ausiliarie che definiscono il tempo al più presto possibile per ogni operazione;

- L'ordine di priorità dei job su ogni centro di lavoro.

3. ***Forward simulation*** . Questa simulazione inizia al tempo corrente e tutti i job devono essee schedulati, inclusi quelli che costtituiscono correnti WIP sulle macchine, sono allocati step by step in ordine cronologico. Nel processo di simulazione si impongono le priorità ottenute dalla simulazione precedemte per selezionare il job da assegnare e si considerano come vincoli le condizioni ausiliarie.

La procedura proposta non è iterativa. Al fine di minimizzare la deviazione dalla due date in una schedulazione basata sulla simulazione è necessario considerare le seguenti funzioni:

- *Il controllo dei tempi di lavorazione* per evitare anticipi eccessivi del job e, allo stesso tempo, la definizione di adeguate regole di priorità per evitare ritardi ne completamento del job;
- *Il bilanciamento dei due interventi* in modo da minimizzare la due-date deviation.

Gli autori propongono un metodo di scheduling caratterizzato da entrambe le funzioni. Per entrambe, inoltre, sono proposti i parametri impiegabili per manipolare il processo di simulazione in relazione alla funzione obiettivo in modo da ottenere una migliore schedulazione. Questa è un'applicazione del metodo di parameter-space-search-improvement (Fuyuki et al. 1998), ovvero la proposta di un nuovo insieme di parametri che sono efficaci nel miglioramento dei risultati dello scheduling per la minimizzazione della deviazione dalla due-date. L'impiego della simulazione permette di realizzare la schedulazione assegnando step-by-step a una macchina, quando questa diventa disponibile, ad un istante fissato qualunque job che rispetti le condizioni operation-onset, ovvero le condizioni, spesso basate sui vincoli di pianificazione come quelli di precedenza, i tempi di setup e di processamento, che devono essere soddisfatte dal job per iniziare la sua operazione sulla macchina assegnata. Le condizioni possono essere, ad esempio, la necessità che la macchina sia libera durante tutto il periodo necessario all'operazione, che i materiali necessari siano disponibili, che le operazioni precedenti sul job siano completate o che sia possibile effettuare il

setup. In più, però si devono considerare ulteriori condizioni che specificano il tempo di realizzazione dell'operazione possibile al più presto e al più tardi. Queste condizioni ausiliarie permettono di contenere l'anticipo del completamento del job. Tuttavia, l'impiego di tali limitazioni risulta piuttosto complesso in relazione ad un numero elevato di job spesso in competizione. Per questo motivo gli autori hanno sviluppato tale approccio alternativo. Al fine di valutare l'efficacia del metodo proposto è stato considerato, oltre ad un problema semplice chiarificativi della procedura, un problema di schedulazione job shop su larga scala considerando oltre cento prodotti, sedici centri di lavoro ciascuno con sei macchine e da tre a dodici operazioni per ogni job. Il caso analizzato, permette, quindi, di considerare un ulteriore impiego della simulazione nell'ambito dello scheduling. Infatti, in questo caso si è considerato un sistema piuttosto complesso di simulazione che però funge da schedulatore e non da semplice strumento di supporto e che ottimizza una funzione obiettivo difficilmente analizzabili con l'impiego di tecniche tradizionale ma di elevata valenza in ambito produttivo.

*L'impiego dei sistemi di simulazione per il controllo del shop flow*

Molti autori hanno analizzato lo sviluppo di modelli di simulazione come strumento di supporto al processo di controllo a livello operativo. In particolare, è stato analizzato l'approccio proposto da Smith et al. [156], che introducono la possibilità di realizzare il controllo di sistemi FMS impiegando la simulazione. Quindi, il simulatore non è solo usato come uno strumento di analisi e valutazione ma anche come un "task generator" per la specificazione dei compiti di controllo. In questo modo, la logica di controllo di sistema non è applicata due volte. Infatti, nel caso della generazione/modifica di un sistema l'approccio tradizionale richiede la valutazione del progetto attraverso la simulazione e la successiva realizzazione/modifica del sistema di controllo. Gli autori propongono un approccio alternativo basato sull'impiego dello stesso sistema di simulazione

utilizzato per la verifica anche per il controllo, che permette di non duplicare gli sforzi di controllo, contenendo in tal modo i costi. L'implementazione è stata effettuata in due laboratori nell'ambito di un progetto congiunto tra la Texas A&M and Penn State Universities e la Systems Modeling Corporation definito come RapidCIM. In sistemi FMS, in particolare, lo shop floor control system è responsabile per la selezione del routing delle parti, operazioni di handling e di scheduling basate sullo stato corrente del sistema. L'approccio proposto sviluppa un sistema di controllo separato in due parti una *esecutiva*, responsabile dell'interazione con l'impianto per l'implementazione dei compiti fisici, e una *decisionale* che si occupa della pianificazione e schedulazione. La separazione è realizzata al fine di ottenere uno sviluppo modulare del sistema di controllo. La struttura del controllore così ottenuto è rappresentata in Figura 2.9. In questa struttura un simulatore Arena/SIMAN [58] funge da decisore o *Task Generator* e un controllore basata su *Message-based PartStateGraph* che descrive il protocollo di processamento, in termini di messaggi ricevuti e spediti e dei compiti che realizza in relazione ad essi, realizza le funzioni esecutive. I due componenti comunicano attraverso un *task initiation queue*, utilizzato dal generatore per definire il compito da realizzare per l'esecutore, e *task completation queue,* impiegato dall'esecutore per comunicare al generatore il completamento del compito, realizzato attraverso l'impiego di controllori locali. In tal modo, si semplifica la struttura del controllore separando le informazioni e le funzioni relative all'ambito decisionale e esecutivo.

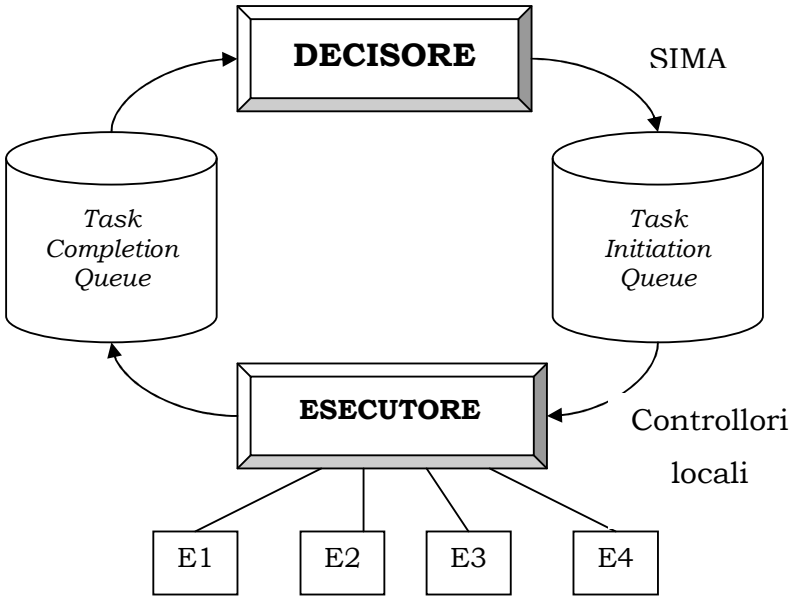



**Figura 2.9 Sistema di controllo operativo proposto**

Per realizzare tale controllore l'Arena/SIMAN è stato in parte migliorato con ulteriori costrutti per facilitare la realizzazione del controllo. Infatti, il tipico approccio di Arena basato sul delay non risulta adeguato in quest'ambito. Tranne che per i componenti del controllo, la struttura del modello Arena è quella tipica dei processi manifatturieri, considerando ad esempio le macchine e i robot come risorse. Dall'analisi sperimentale nei due laboratori in cui il sistema è correntemente in uso si evince che tale sistema di controllo presenta prestazioni soddisfacenti e permette di prevedere le performance future del sistema sulla base delle correnti decisioni.

In particolare, è stato analizzato l'approccio proposto da Smith et al. in quanto dimostra la rilevanza della simulazione anche nell'ambito del controllo soprattutto nel caso in cui l'ambiente produttivo sia molto flessibile e ad alto contenuto tecnologico, come nel caso dei FMS, in cui il controllo assume un'elevata rilevanza.

In genere, nell'ambito del controllo, l'impiego della simulazione come supporto solo ad un livello operativo risulta poco efficace. Infatti, le decisioni

di controllo a questo livello sono spesso interconnesse con quelle strategiche e tattiche. Ad esempio, i target di performance in un processo di controllo sono in genere basati su dati storici e in alcuni casi aggiustati con fattori di miglioramento. Tuttavia, nonostante tali interventi i target non rappresentano la situazione reale del sistema, per cui essendo usati come obiettivi causano un perpetuarsi dell'inefficienza del sistema fino al livello operativo. Per questo motivo, Roy [141] ha proposto l'impiego della DES come strumento di supporto decisionale, che permette di analizzare il sistema in modo da definire target che siano effettivamente rappresentativi e costituisce una strumento trasversale di supporto che permette un'integrazione dei diversi livelli di controllo. In particolare la DES permtte ai livelli più bassi del controllo gerarchico, ovvero il controllo budgetario e quello strategico, realizzati in genere su base annuale, di definire dei target efficaci e sulla base di tali target, nell'ambito del livello di controllo del sistema di produzione e di shop floor control, permette attraverso la simulazione di effettuare la schedulazione del processo e di valutarne gli effetti per definire eventuali intervento migliorativi.

### *2.8.6. I sistemi di simulazione integrati*

Per quanto concerne gli approcci integrati, è possibile considerare differenti applicazioni. In generale, una definizione univoca di questo tipo di sistemi non è possibile in relazione all'eterogeneità degli scopi per i quali sono sviluppati e delle strutture ottenute. In alcuni studi, i sistemi integrati sono stati impiegati a livello strategico per la definizione delle diverse configurazioni del sistema. In questo caso, il modello di simulazione interagisce in maniera dinamica con un sistema di raccolta dati al fine di generare una struttura self-building. Tale approccio risulta di poco differente rispetto ai casi precedentemente analizzati in ambito strategico; tuttavia permette di contenere gli elevati costi dovuti in particolare alla raccolta dei dati, allo sviluppo del modello e alla sua validazione. Per questo

motivo, è possibile impiegare il sistema in ambiti, quali le piccole imprese, in cui in genere il Business Process Engineering non è per questi motivi supportato dalla simulazione. Questo tipo di analisi è stata effettuata in maniera esaustiva da Mosca et al.[124] nell'ambito del flow shop impiegando come software di simulazione Arena 8.0. Il caso esaminato per la valutazione sperimentale dell'analisi effettuata è quello di un'azienda di piccole dimensioni con sette operai e otto centri di lavorazione. In particolare, lo sviluppo del sistema è stato realizzato in tre fasi:

1. **Data collection,** che permette di raccogliere le informazoni relative ai flussi di lavoro e ai tempi di processamento individuando le distribuzioni statistiche più appropriate attraverso le tecniche di valutazione della goodness-of-fit;
2. **Implementazione del modello** che comprende anche l'integrazione con database adeguati per supportare la validazione e la verifica. Iniziando da un piano della produzione simulato, un ordine di produzione è generato, diviso in una distinta base caratterizzata da quattro livelli, creando un ordine per ogni sottoprodotto e assegnando a questo un valore di priorità. Quindi, l'ordine è suddiviso in diversi sottordini rappresentanti ciascuna entità da produrre. Quando un'entità è processata può liberare il suo livello superiore che procede nel sistema, oppure essere unita, il che rappresenta l'assemblaggio, con un'altra entità e poi il nuovo assemblato libera il livello superiore o, infine, uscire dal sistema aggiornando il database;
3. **Valutazione** della relazione funzionale tra diversi fattori per selezionare la migliore configurazione.

Il modello di simulazione è integrato con un database relazionale (MS Access 2003) costruito da diverse tabelle. Le prime due contengono i dati relativi alle date, l'utilizzatore può gestire le eccezioni usando lo schedulatore di Arena, e gli ordini pianificati, in termini di due date, idorder, idproduct, quantità richiesta e priorità. Un'altra tabella le distinte base multilivello e multiprodotto. Infine, il database funge da magazzino virtuale da quelli che sono prodotti spediti sottratti e elementi prodotti aggiunti (parti assemblate o

prodotti semi-finiti).Inoltre, c'è una tabella di collegamento tra database e modello, che contiene gli attributi SIMAN per ogni entità. L'ultima tabella è usata per la raccolta dei dati.

Il modello sviluppato dagli autori presenta un modulo VBA separato connesso con SIMAN in cui quattro cicli sono innestati per la generazione degli ordini. Quando l'ordine è generato due attributi sono associati all'entità: la quantità da produrre e quella da mantenere. I prodotti con maggiore priorità, il che implica valore aggiunto più elevato, sono assemblati e spediti il più presto possibile mentre gli altri sono processati quando le risorse sono disponibili. Un algoritmo interno basato sul MRP schedula i sottoassiemi e le parti secondo la loro due date finale. Dopo che l'ordine ha generato le diverse entità con i diversi attributi, queste sono in parte spedite ad un magazzino logico e in parte al primo modulo della loro sequenza di lavorazione. Le sequenze realizzate sono 52 e il percorso da seguire è mantenuto dall'entità come un attributo. Quando un'entità deve essere prodotto è divisa in due entità, la prima prova a prelevare il prodotto dalle scorte, la seconda segue la sequenza di lavorazione, poi le entità si ricongiungono. A questo punto l'entità è soggetta a due scelte a seconda della priorità, ovvero può essere immagazzinata o impiegata per realizzare il prodotto padre. In quest'ultimo caso, può essere congiunta con un'altra entità. Ogni macchina è modellata utilizzando regole di pre-emption o di dimensionamento a seconda delle necessità. La lunghezza della simulazione è stata valutata utilizzando la curva a ginocchio del Mean Square Pure Error per valutare lo stato stazionario del sistema. La funzione obiettivo sviluppata considera il valore prodotto e i WIP pesati con la priorità del lotto, la linea usata e il tipo di prodotto. Fattori di penalità sono stati introdotti per considerare le risorse umane assunte, le stazioni totali di assemblaggio disponibili e quelle di affilatura. Altre penalità sono utilizzate per valutare il non rispetto dei target misurati come il rapporto il valore dei prodotti producibili e quello di produzione ottenuto. Infine, si considerano dei coefficienti per considerare la penalità associata al ritardo. Questa funzione è stata implementata in un modulo post processing e analizzata utilizzando il

DOE e le Superfici di Risposta nella fase di valutazione delle configurazioni. I risultati di tale fase, sviluppata considerando come fattori il numero di centri di assemblaggio (2-4) la capacità di lavoro delle macchine di affilatura(2-4), il volume totale di produzione (0,8-1,2) e il numero di risorse umane(7-9), sono stati elaborati attraverso un programma Excel in VBA. La struttura globale ottenuta si è dimostrata essere particolarmente flessibile. Quindi, gli autori presentano una struttura di simulazione genral-purpose che, impiegata in maniera congiunta con strumenti di analisi statistica, permette di valutare la relazione tra diversi fattori selezionati e le prestazioni al fine di selezionare la configurazione migliore.

Per quanto concerne i sistemi integrati sviluppati a livello tattico e operativo, invece, è presente in letteratura un numero più ampio di studi, anche in questo caso molto differenziati. Un primo approccio considera semplici integrazioni del modello di simulazione con il sistema Enterprise Resource Planning (ERP), al fine di migliorarne l'efficacia e le prestazioni. Pur basandosi tutte sulla considerazione che la simulazione debba essere integrata con un sistema ERP per permettere una valutazione più dettagliata e realistica, le strutture proposte sono abbastanza eterogenee in termini di modalità di integrazione e di sviluppo del modello di simulazione, come si dimostra dall'analisi del sistema proposto da Musselman et al. [183] e di quello, basato sullo sviluppo del software SIMUL8-Planner, realizzato da Concannon et al.[47].

Nel primo caso [183], gli autori descrivono una funzione di scheduling simulation-based integrata con un sistema ERP. Moderni sistemi ERP contengono tutti i dati necessari per la pianificazione dettagliata della produzione. Questi includono informazioni sul prodotto, come distinta base e routing delle parti; informazioni sul sistema, quali orza lavoro e impianti e informazioni sullo status come ordini attuali, WIP, livelli di scorte e ordini di acquisto rilasciati. Questi dati sono necessari alla funzione di Advanced Planning and Scheduling (APS) per determinare come pianificare le operazioni. Molti studi sono stati effettuati al fine di effettuare applicazioni che potessero evidenziare eventuali irrealizzabilità del MRP con

schedulazioni più realistiche basate sulla simulazione. Lo studio proposto presenta un approccio più ampio basato sull'impiego dell'APS. In particolare effettua la pianificazione tenendo conto dei vincoli di capacità e realizza un piano schedulabile , a differenza del MRP, sulla base del routing delle parti, della distinta base, del livello di scorte e della domanda comprensiva di previsioni e scorte di sicurezza e della capacità disponibile. La pianificazione è effettuata prima backward dalla due date. In tal modo, se la data di inizio è in passato riorganizza la produzione, se comunque non si soddisfa la due date l'ordine è eliminato. Il piano di massima, che definisce le date di rilascio e completamento per ogni domanda, è schedulato in modo da ottenere una lista delle operazioni e di come la capacità è impiegata. Tali informazioni sono analizzate per aggiornare il piano e anche per valutare la possibilità di soddisfare un ordine di un nuovo cliente. L'advanced scheduler determina quale job lavorare su ciascuna stazione e quando, con un orizzonte temporale giornaliero o settimanale. La schedulazione considera i run time variabili in funzione della stazione e dell'operatore, le regole di assegnazione delle macchine basate anche sui requisiti di qualità, i tempi di setup variabili e tra gli obiettivi i costi e le due date. L'articolo impiega il Frontstep APS integrato con il syteLine del Frontstep e il sistema System21 ERP di GEAC. La sua funzione è il coordinamento della pianificazione dei materiali e della capacità per soddisfare la domanda e si basa sui tre processi di advanced planning, advanced scheduling e order promising. Il ruolo della simulazione nel sistema APS è critico in relazione alla possibilità di ottenere una rappresentazione più fedele della realtà. Tale valore è ancora maggiore in un tale ambiente integrato.

Concannon et al. [47], invece, hanno sviluppato un'applicazione software SIMUL8-Planner, realizzata da SIMUL8 Software e Visual8 Corporation, come strumento di supporto alla pianificazione e alla schedulazione. In particolare, gli autori considerano l'approccio simulativo come un mezzo di realizzazione di piani di produzione consistenti e corretti che integrandosi anche con i sistemi ERP aziendali, permettono di compensarne l'incapacità di adeguarsi in maniera rapida e efficiente ai cambiamenti, soprattutto in

termini di situazioni non previste quali breakdown delle macchine, carenza di materiali o di risorse, a causa di un livello di dettaglio non elevato. Il Simul8-Planner si collega direttamente all'ERP del sistema per ottenere informazioni relative ai processi di produzione, agli ordini, alle distinte base e ai dati relativi alle scorte per supportare le decisioni di scheduling . Sulla base dei vincoli di capacità e degli obiettivi di gestione, si genera un piano di produzione pre-simulazione. Questo confronta gli ordini di produzione con le scorte di prodotti finiti e definisce le esigenze di produzione nel tempo. Se gli ordini eccedono la capacità disponibile sono assegnati a periodi precedenti. Questa pianificazione, tuttavia, non assicura in alcun modo che gli ordini siano completati senza ritardo. Il piano è basato sulle decisioni e regole di scheduling contenute in euristici interni ed è eseguito da un sistema di scheduling basato sulla simulazione che genera un efficiente e realizzabile scheduling di produzione, considerando vincoli di capacità, regole di priorità e target di produzione. I vantaggi di tale approccio rispetto ai tradizionali metodi statici è il collegamento diretto e dinamico con lo shop floor. Inoltre, le soluzioni possono essere costumerizzate a seconda della pianificazione corrente e delle regole di scheduling. Queste ultime sono sviluppate con la simulazione sono flessibili e facilmente modificabili.

La simulazione permette di valutare e adeguare la pianificazione a rapidi cambiamenti del processo, come il breakdown delle macchine e la disponibilità della forza lavoro in maniera rapida, adifferenza dell'ERP. L'approccio simulato fornisce anche i mezzi necessari all'assestamento dell'efficacia della schedulazione valutando i risultati, come l'utilizzazione delle macchine, gli straordinari necessari e altre analisi standard dei risultati Invece, i metodi euristici tradizionali permettono di effettuare tale valutazione solo attraverso l'implementazione del piano nel sistema reale, limitando la possibilità di anticipare eventuali cause di inefficienza e irrealizzabilità. Inoltre, analisi what-if multiple possono essere effettuate per assicurare che i piani realizzati siano coordinati con le correnti necessità manageriali. I principali tre componenti dell'applicativo analizzato sono:

- **Strumenti di gestione del tempo** per determinare gli slot di tempo disponibili per l'allocazione dei job sulle macchine. Infatti, è possibile utilizzare specifici calendari per definire i tempi di riparazione e di guasto per le singole macchine, in modo da realizzare schedulazioni individuali, e i tempi di inattività dell'intero impianto.
- **Tavole indicizzate e routine di ricerca** per ricercare e manipolare i dati della simulazione, come gli ordini di produzione, i file relativi alle scorte. Le routine permettono di ricercare un ordine specifico dalle tabelle, aggiungere o eliminare gli ordini e individuare alcuni criteri come le due date. Molte di queste funzioni sono simili ale queries usate dai sistemi ERP;
- **Gantt interattivi** che mostrano la schedulazione della produzione. Da tali diagrammi è possibile anche estrapolare informazioni sul singolo job in termini di tempi di inizio lavorazione e fine, inserire testi relativi ai job e differenziarli utilizzando diversi colori. Inoltre, è possibile realizzare funzioni custmizzati per la rischedulazione, per ottenere report complessivi. Infine, è possibile modificare il diagramma e rieffettuare la simulazione per valutare l'impatto di tali modifiche nello scheduling sul sistema. In questi casi, è possibile definire diversi accessi per gli utenti.

Infine, il sistema può essere connesso con altre applicazioni, come ad esempio i sistemi di Manufacturing Intelligence (MIS) per aggiornare i dati in tempo reale e aggiornamenti sullo stato del sistema. La struttura ottenuta permette, quindi, una schedulazione flessibile e reattiva ai cambiamenti attraverso la sincronizzazione con i sistemi di manici di produzione.

Da semplici integrazioni con l'ERP, si passa a strutture poco complesse che, però, integrano il modello di simulazione con sistemi di generazione automatica del piano di produzione e di schedulazione per migliorarne le performance, come quella proposta da Marvel et al.[114]. In questo caso, l'integrazione riduce i tempi e i costi di pianificazione. Gli autori considerano l'integrazione della DES nell'ambito della pianificazione della capacità, utilizzando come caso studio un'industria di prodotti metallurgici. L'obiettivo

della pianificazione della capacità è di determinare la sequenza di prodotti su ciascuna linea e la quantità di produzione necessaria alla soddisfazione del cliente. Il tradizionale processo di pianificazione della capacità ha inizio con la pianificazione aggregata della produzione e delle risorse necessarie che permette la generazione del Master Production Scheduling e del Rough- Cut Capacity Planning. A questo punto, è stata esaminata solo la capacità dei centri critici. Dopo l'esplosione della distinta base si genera il Material Requirement Planning (MRP) e si realizza la schedulazione. Oltre alle limitazioni già discusse per il MRP, tale approccio non considera adeguatamente l'incertezza sulla domanda e la disponibilità effettiva degli impianti. Nel caso esaminato il piano di capacità determina quali prodotti devono essere schedulati su un ciclo ripetitivo e su quale linea. I prodotti non schedulati esplicitamente sono realizzati nei gap sulle differenti linee. Inoltre, la pianificazione, nel caso specifico, considera la disponibilità delle bobine che sono fornite dal cliente e il cui quantitativo è negoziato all'inizio del lancio dell'ordine. Il prodotto realizzato è, quindi, posizionato sulle bobine e inviato al cliente. Il cliente, poi, invia le bobine vuote al processo di produzione. Le bobine sono spesso specifiche per i prodotti. Il sistema ha stoccati un numero limitato di bobine per compensare eventuali sottoscorte di bobine. Dall'analisi si evince che il processo può essere schematizzato come una rete di code chiusa. L'obiettivo della simulazione non è solo quello di validare il piano di capacità tenendo conto dei livelli di scorta e di soddisfare la domanda del cliente ma anche quello di progettare il numero specifico di bobine del cliente necessarie. L'analisi è effettuata su un orizzonte temporale di 3-12 mesi. Inoltre, l'identificazione dei prodotti con maggior volumi è stata effettuata con un analisi di Pareto. Attraverso il modello giornaliero delle spedizioni è stato realizzato il kanban giornaliero, che indica la domanda e non la produzione. Quindi, è stata pianificata la sequenza e l'assegnamento dei prodotti schedulati sulle linee. Tuttavia, la pianificazione della capacità non permette di capire se ci sono abbastanza bobine per soddisfare la domanda e abbastanza slot di capacità per i prodotti non schedulati. A questo punto, per sopperire a tali limitazioni è stato

realizzato un modello di simulazione che traccia le bobine e i prodotti attraverso il sistema. Il modello è costituito da cinque moduli in input:

- Capacity planning output che fornisce informazioni relative alla sequenza dei prodotti e alla schedulazione. Queste informazioni sono impiegate per determinare gli slots di capacità;
- Dati specifici del prodotto, come ad esempio i tempi di setup e cambio delle bobine, tassi di produzione, associazione bobine/cliente;
- Dati di scorte delle bobine, la localizzazione delle bobine nel processo, se sono piene o vuote;
- Dati di spedizione dal MRP relativi al quantitativo di prodotti spediti al cliente ;
- Dati specifici dell'utente in quanto l'utente può customerizzare l'esperimento di simulazione cambiando alcuni requisiti del modello, come la percentuale di bobine richieste per iniziare la produzione.

Il flusso informativo nel sistema è mostrato in Figura 2.10.

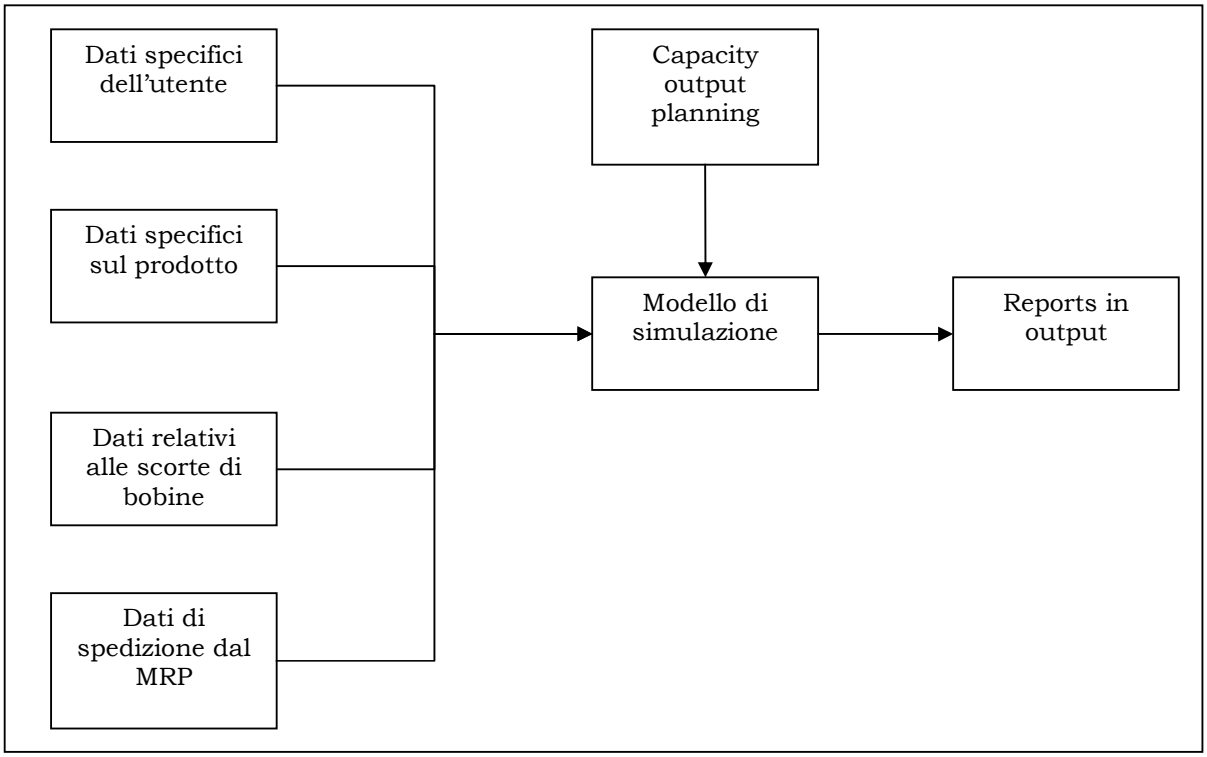



**Figura 2.10 Flusso informativo**

La simulazione effettua il processamento di tutti i prodotti schedulati. Se non è possibile realizzarli a causa di carenza di bobine il sistema determina se utlizzare le bobine di riserva o attendere l'arrivo di bobine dal cliente in base ai dati a disposizione. Dopo la schedulazione di tali prodotti il sistema valuta se ci sono slots disponibili e li schedula sulla base delle relative regole di priorità. Il sistema permette anche di considerare la gestione del magazzino sulla base degli ordini futuri, di produrre senza rimandare nel caso di carenza di bobine anche usando quelle di un altro cliente i backorder e di definire gli straordinari necessari per la realizzazione dei prodotti non schedulati. Il modello di simulazione è stato sviluppato utilizzando il software Promodel e i risultati sono stati esportati su Microsoft Excel. Gli output di simulazione sono utilizzati per valutare se il piano di capacità e la sequenza dei prodotti schedulati è realizzabile. Inoltre, l'utente può valutare la presenza di backorders in funzione della carenza di bobine o di insufficienti slots di capacità. Inoltre, è possibile modificare alcuni parametri per effettuare delle analisi "what-if".

Il modello di simulazione così sviluppato è uno strumento che permette non solo di validare la pianificazione della capacità ma anche di schedulare il bilanciamento delle linee di produzione, valutare i problemi connessi alle bobine e agli ordini non schedulati e valutare diverse configurazioni possibili in un'ottica di miglioramento continuo. Gli autori, quindi, integrano il sistema di simulazione nel processo di pianificazione della capacità per verificare la fattibilità del piano, schedulare i prodotti che il processo di pianificazione non riesce a gestire, di pianificare la fornitura dei materiali dai clienti e di identificare le aree di miglioramento della produzione al fine di definire gli interventi migliorativi da realizzare.

Infine, sono stati sviluppati sistemi più completi come quelli proposti da Metal et al.[118] e da Kuhen et al.[103] per processi di tipo job shop. Entrambi gli approcci effettuano la realizzazione in maniera automatica della schedulazione; nel primo caso integrandola con sistemi di controllo di processo e alberi di learning, nel secondo attraverso la generazione

automatica del modello realizzata mediante un'interfaccia con un database e l'integrazione con il sistema di pianificazione.

Per quanto concerne Metan et al. [118], presentano un meccanismo di learning basato sull'integrazione del modello di simulazione con l'ambiente circostante. Il sistema impara nell'ambiente produttivo costruendo un albero di learning e seleziona la dispatching rule dall'albero per ogni periodo di scheduling. Il sistema utilizza il diagrammi di controllo di processo per monitorare le performance dell'albero di learning che è automaticamente aggiornato se necessario, in tal modo si adatta ai cambiamenti dell'ambiente produttivo. Gli autori hanno sviluppato tale sistema in quanto i modelli di simulazione in genere impiegati non realizzano lo scheduling in maniera integrata con il controllo. Per questo motivo, seppur permettendo attraverso la simulazione un'analisi più prossima alle reali condizioni del sistema, il modello non è automaticamente aggiornato. In particolare, analizzando attraverso la simulazione le performance delle diverse regole di dispatching per la schedulazione, nessuna di queste si è dimostrata superiore in ogni condizione. Inoltre, si è dimostrato che l'impiego di diverse regole, ovvero di metodi multi-pass, risulta migliore rispetto all'utilizzo di una sola, metodi single-pass, per l'intero orizzonte temporale. In genere, in quest'ultimo caso si simula un insieme di regole e si seleziona quella con migliori performance nel lungo periodo. Nel caso multi-pass, invece, per ogni intervallo ridotto si impiega la regola che risulta migliore. Quindi, nel lungo periodo il processo risulta una combinazione di diverse regole. Tuttavia, tale approccio richiede un tempo di simulazione elevato per valutare le performance in ogni intervallo di ogni regola. Inoltre, si suppone nota la distribuzione di probabilità e i parametri del tempo di processamento e di arrivo. Nel caso di tipologie di prodotti che cambiano spesso, come quelli ad alto contenuto tecnologico, questo non si verifica in quanto i tempi di processamento possono variare a causa dell'obsolescenza delle macchine. L'articolo propone un sistema di selezione delle regole che usa le tecniche di intelligent machine learning e le carte di controllo. Il sistema è illustrato in Figura 2.11.

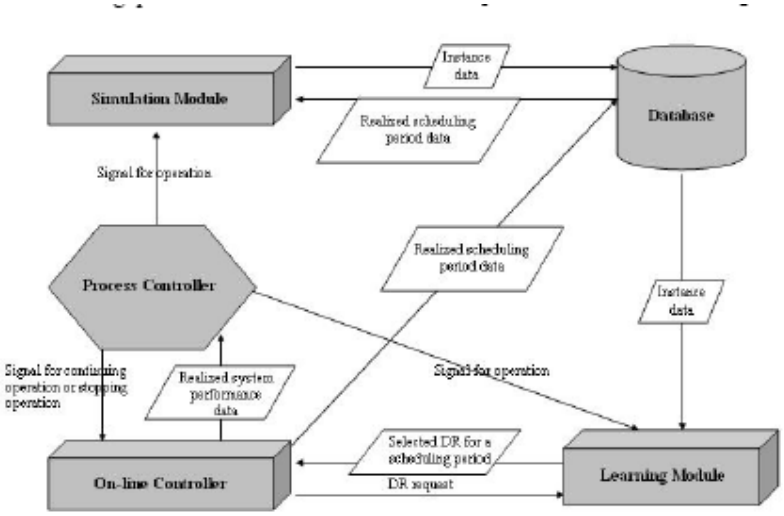



**Figura 2.11**

Il database fornisce i dati necessari per il sistema di learning e quello di simulazione. Questo contiene gli instance data che sono i dati relativi alle condizioni del sistema e alla regola corrispondente selezionata. I dati di scheduling di periodo realizzato, invece, rappresentano gli eventi attuali che si verificano in uno specifico periodo di scheduling come tempi di processamento, intertempi di arrivo e condizioni del sistema all'inizio del periodo di scheduling. Il modello di simulazione misura le performance delle regole candidate ed è richiamato da un modulo controllore di processo se necessario. Gli output del modulo di simulazione sono inviati al database. Questi risultati sono utilizzati dal modulo di learning per generare l'albero. Il modulo di learning è costituito da due parti, la prima contiene l'albero, la seconda l'algoritmo di costruzione dell'albero stesso. Il modulo di controllo on-line, invece, fornisce i valori correnti degli attributi dello stato del sistema al modulo di learning che determina la regola sulla base dell'albero esistente e la invia al controllo che la implementa. Il controllo, inoltre, fornisce i dati relativi allo scheduling realizzato nel periodo al database e monitora il sistema reale per la selezione della nuova regola e gli eventi che permettono di definire quando schedulare. Il modulo di controllo di processo monitora le performance dell'albero sulla base degli input, in particolare in termini di

ritardo medio, dal controllo online. La valutazione è realizzata sulla base di carte X medio e R di controllo. Se queste carte presentano punti esterni alle linee di controllo, o se per la X medio di hanno 2 o 3 punti tra 2σ e 3σ o sotto o ci sono 4 o 5 punti tra σ e 2σ o sotto o se si è all'inzio di un periodo di schedulazione si aggiorna l'albero. Inoltre, se si hanno due segnali successivi di aggiornamento dell'albero o 8 punti successivi strettamente sotto o sopra la linea centrale per entrambe le carte si aggiorna il processo di controllo. Quando le performance dell'albero sono non soddisfacenti richiede al simulatore di fornire nuovi instance data per il modulo di learning e manda un segnale a tale modulo perché aggiorni l'albero. Quindi, nuove regole sono generate sulla base dell'albero aggiornato.

È stato considerato, come caso sperimentale, il problema di un job shop per analizzare il periodo di schedulazione e la distanza di due punti monitorati sulla base di tre funzioni di performance, Il multi-pass ovvero il ritardo medio ottenuto da un simulatore con scheduling multi-pass, il best performance ovvero il valore minimo di ritardo medio possibile per un periodo utilizzando ciascuna regola candidata e il learning performance, ovvero il ritardo medio realizzato dalla regola scelta dall'albero. Il piano fattoriale è caratterizzato da due fattori, due date (persa o soddisfatta) e utilizzazione (bassa o alta) con 20 replicazioni e analisi allo stato stazionario. Per la lunghezza del periodo si verifica che il Bestperf è peggiore rispetto a quelli tradizionali, in particolare al single-pass, perché il continuo switching ne altera le performance di breve periodo. Aumentando la lunghezza si raggiunge un minimo del ritardo che poi però riprende a crescere. Per quanto concerne la distanza tra i unti, si verifica che le performance del sistema peggiorano al livello delle performance del multi-pass.Il learning system è stato valutato considerando un sistema di schedulazione multi-pass reattivo, parzialmente reattivo e no reattivo. Tale classificazione è effettuata sulla base di quanto rapidamente si aggiorna il sistema al variare dei parametri, quali i tempi di arrivo. In particolare si focalizza l'analisi sul sistema più realistico, ovvero quello parzialmente reattivo che si aggiorna con qualche ritardo. In questo caso, learnperf è migliore del Multipass,

ovvero più prossima alla bestperf in gran parte dei casi o al massimo sono molto prossime.

L'ultimo studio proposto, che fornisce probabilmente il più completo sistema integrato, è quello proposto da Kuhen et al. [103]. In questo caso, gli autori sviluppano un *Simulation Based Job Shop Analyser* come strumento di analisi a supporto della pianificazione e controllo operativi che integra l'impiego di applicazioni Java e di database. Per la painificazione e il controllo può essere impiegato un sistema di dispatching rule based (RBD) per valutare la schedulazione dei job sulle macchine. Il simulatiom Based analyser offre la possibilità di un approccio flessibile e efficiente per simulare in maniera rapida il sistema di produzione di elevate dimensioni di tipo job shop. Tale strumento offre i seguenti vantaggi:

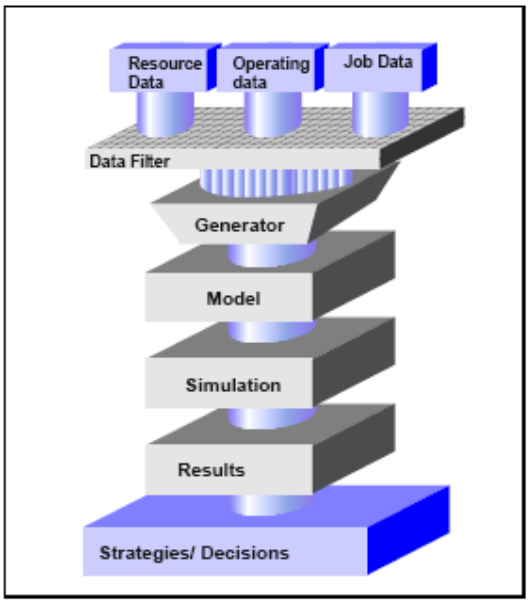
- Generazione automatica del modello;
- Simulazione integrata;
- Interfaccia con il database;
- Effettuazione in tempi ridotti della simulazione;
- Integrazione del sistema nell'ambiente produttivo;
- Possibilità di customizzazione.

Il sistema può essere integrato nella rete esistente per la pianificazione e il controllo. I dati in input possono essere ottenuti da un interfaccia con il database o XML. I dati richiesti in input possono essere classificati in tre categorie:

- Dati relativi alle risorse: Parametri di produzione, macchine, centri di lavoro;
- Dati operativi: assegnazione dei calendari, dati di produzione;
- Dati relativi ai job: lista dei job, date e priorità.

I dati relativi alle risorse sono statici, ovvero in genere non cambiano se non attraverso modifiche sostanziali del sistema. I dati operativi, invece, si modificano nel caso di interventi sul processo. Infine, i dati relativi ai job cambiano con frequenza giornaliera. I dati sono impiegati per la generazione automatica del modello. In ogni caso il modello è automaticamente generato dal database. In questo modo, non ci sono discrepanze tra i dati impiegati

dal modello e quelli nel database. Se necessario, alcune aree specifiche possono essere modellate più in dettaglio utilizzando specifici building blocks.




L'architettura software dell'ultima versione è basata sull'impiego di linguaggi Java. I risultati sono riportati sul database. In ogni caso, l'analisi dei risulatati può essere effettuata direttamente impiegando le interfacce utente dal modello o rilevando i dati dal database.Il simulatore può essere pienamente integrato con il sistema informativo esistente, ad esempio le interfacce utente per l'immissione di dati e l'analisi dei risultati possono essere collocate su un qualunque computer della rete. Inoltre, l'impiego della libreria Java permette la definizione delle caratteristiche del sistema in maniera molto semplice. Tale architettura, che impiega building bloks come le stazioni di lavoro, di assemblaggio, è basata su un approccio object oriented modulare per permettere ulteriori estensioni. Questo strumento permette la modellazione e simulazione di ogni tipo di produzione job shop. Ad esempio, è stato sviluppato nell'ambito dei semiconduttori in quanto il processo di realizzazione risulta molto complesso, richiedendo più di 100

centri di lavoro con circa 350 macchine, oltre 1000 prodotti diversi e la presenza di ordini urgenti. Il sistema permette, inoltre, di effettuare la schedulazione considerando più 50 differenti regole .Infine, un'analisi dei risultati molto ampia che considera numerosi fattori di influenza come compiti addizionali, cambiamento nelle regole di schedulazione e nella dimensione dei lotti, cambiamenti nella struttura e numerosi parametri di performance quali il numero di prodotti finiti, i WIP, utilizzazione delle risorse e misure definite dal cliente può essere realizzata. Quindi, questo strumento offre la possibilità di analizzare diverse strategie relative alla gestione operativa della produzione in maniera molto flessibile.

Dall'analisi effettuata, si verifica che anche nel caso dei sistemi integrati gli approcci sviluppati sono fortemente particolarizzati sulla base delle esigenze dello specifico processo produttivo preso i considerazione. Inoltre, il numero di sistemi globalmente integrati e non solo parzialmente con alcuni elementi dell'ambiente produttivo è molto ridotto. Tuttavia, numerosi sono gli sforzi orientati alla generazione di questo tipo di struttura in relazione ai numerosi vantaggi in termini di efficacia, dinamicità, flessibilità ed efficienza ottenibili.

In quest'ambito sono stati considerati solo sistemi integrati per un singolo ambiente produttivo. Tuttavia, sulla base di tali possibili configurazioni sempre maggiore attenzione è posta all'integrazione di più ambienti attraverso l'impiego della simulazione. Ne è un esempio l'approccio proposto da Ruiz-Torres e Nakatani [142], che hanno sviluppato elementi integrati di fornitura e trasporto basati sulla simulazione per l'assegnazione di due date nell'ambito delle reti logistiche e manifatturiere.

## 2.9. Considerazioni e conclusioni

Malgrado il fatto che sia stata effettuata una approfondita analisi comparativa riguardo alle tecniche di soluzione di $\Pi_j$, c'è un forte dibattito

per quanto concerne il come i risultati dei vari metodi, specialmente quelli euristici, potrebbero essere presentati. Attualmente sono previste solo poche linee-guida e, di conseguenza, non sono disponibili procedure standard. Non sorprende che in molti casi non è chiara l'interpretazione che può essere data dei risultati. In modo da dimostrare e superare questo gap, Barre et al. (1995) sottolineano alcune delle questioni di progettazione e presentazione computazionale degli esperimenti per test euristici. Questi studiosi hanno concluso che schematizzazioni degli studi sperimentali sono ancora in una fase primordiale e c'è la necessità di una maggiore archiviazione dei diversi problemi, dei test bed e dei codici. Hooker (1995) consiglia un lavoro più scientifico da applicare che si enfatizzi sulla ricerca di una misura per valutare tali problemi. Anche Reeves (1993) crede che sia richiesta una più rigorosa progettazione sperimentale per la scelta di una euristica relativa ad un dato problema, in quanto i motivi della scelta non sono chiari e comunque dipendono da diversi fattori.

Questi sono solo alcuni punti di considerazione ma ci sono altri aspetti che meritano di essere considerati. Sebbene da un punto di vista empirico, gli approcci di ricerca locale hanno mostrato di essere il più appropriato algoritmo di approssimazione, specialmente per i problemi che incrementano di dimensionalità, studi teorici hanno caratterizzato il fallimento di questi metodi, specialmente per un caso di prospettive peggiori. Per questo è necessario provare e capire la teoria della prestazione tempo-finita della ricerca locale e quando e quali approcci euristici hanno possibilità di successo o di fallimento. Altre questioni non risolte sono quelle relative alle proprietà di convergenza, alla capacità di garantire che le soluzioni sono ottime e alla complessità della ricerca di un ottimo locale. Attualmente non è possibile dare un limite non banale sul numero di iterazioni necessarie per trovare un minimo locale.

C'è anche un metodo non formale per suggerire l'effettiva strada nella combinazione delle tecniche meta-euristiche. Per esempio Yamada e Nakano (1995, 1996) credono che la più appropriata inserzione per SBP nell'algoritmo SA è durante la fase di accettazione/rifiuto. Uno studio

potrebbe concentrarsi su come combinare i vari metodi oppure su come separare un modello ibrido e ricombinarlo in modo da farlo diventare un più potente e appropriato strumento di ricerca.

Data la natura combinatoria di $\Pi_j$ i metodi attuali hanno estrema difficoltà nel ricercare effettivamente spazi di soluzione per problemi di dimensionalità 15. Possono essere applicati simultaneamente approcci paralleli per ricerche multiple dell'intorno in modo da migliorare le tecniche esistenti. Sfortunatamente il più recente lavoro su tale approccio ha riscontrato un limitato successo (Taillard 1994, Perregaard e Clausen 1995, Boyd e Burlingame 1996, Ten Eikelder et al. 1997 e Resende 1997).

Si considera che i metodi AI abbiano già mostrato il loro vero potenziale. Approcci NN correnti, applicati a $\Pi_j$ sono sfruttabili solo per pochi problemi ed hanno tempi di calcolo molto elevati. Tuttavia NN diventa utile nel caso si compongano problemi ibridi. Applicando opportune tecniche un vasto spazio di ricerca può essere ridotto e si può permettere l'impiego di meta-strategie per ricercare la schedulazione ottimale in uno spazio decisionale ridotto.

Un'altra area che merita una analisi maggiore è quella relativa ai metodi di approssimazione e ottimizzazione tali che la convergenza a LBs ottimale è conseguita tramite la parte dell'algoritmo esatta, usando un UB ottenuto con la parte euristica. I risultati ottenuti da Caseau e Laburthe (1995), Thomsen (1997) e Nuijten Le Pape (1998), indicano il loro potenziale.

L'applicazione di metodi bottleneck, come l'SBI (Adam set al. 1988) e di inserzione, potrebbe aiutare a conseguire rapidamente schedulazioni iniziali di alta qualità.

In aggiunta, come potenziamento, potrebbero essere adottate per selezionare i movimenti, alcune strutture di intorni, come ad esempio quelle a blocchi critici (Vaessens et al. 1995). Le soluzioni si possono migliorare con una probabilità di accettazione che decresce nel tempo (come negli algoritmi soglia), in cui verso la fine della ricerca sono accettati solo valori migliorativi. Inoltre, potrebbe essere usato un metodo deterministico piuttosto che random per selezionare le nuove mosse.

I sistemi, così, genererebbero solo schedulazioni attive in modo da garantire che è stato ricercato il più piccolo spazio di soluzione contenente la soluzione ottima. Baker (1974) indica che con un obiettivo di $C_{max}$, le tecniche euristiche che producono schedulazioni semi-attive, sono migliori di quelle che producono schedulazioni attive. Di conseguenza molte tecniche generano sequenze semi-attive (Brucker et al. 1994, Mattfeld 1996, Nowicki e Smutnicki 1996, Sabuncuoglu e Bayiz 1997). Ciò è dovuto al fatto che questi metodi non hanno o hanno pochissime strategie di diversificazione e per questo sono solo capaci di effettuare una intensificazione della ricerca in un intorno della soluzione iniziale o di elite. Se l'ottimo globale è in questa area può essere trovato, ma per molti problemi questo non avviene. La conseguenza è che alcune strategie di diversificazione potrebbero essere applicate per esplorare regioni dello spazio di soluzione, inesplorate. La diversificazione può essere applicata anche ai minimi locali per ottenere schedulazioni che abbiano risultati migliori.

L'analisi indica che la ricerca su $\Pi_j$ è stata estesa ed ha compreso un ampio spettro di approcci. Per ottenere dei progressi è necessario che gruppi di lavoro legati ad approcci completamente diversi si applichino insieme per la risoluzione del problema. E' indispensabile costruire un ambiente che racchiuda un insieme di diverse idee per la creazione di nuovi concetti.

Sebbene siano stati fatti notevoli progressi negli ultimi anni, una difficoltà inerente tutti questi metodi è quella delle mancanza di un algoritmo euristico che abbia una prestazione garantita. Di conseguenza per molti metodi approssimati, esistono problemi per i quali questi non processano in maniera adeguata o completa e sono relazionati a diverse circostanze che possono generare successi o fallimenti del processo. E' quindi evidente che, affinché le barriere attuali al problema $\Pi_j$ siano superate, è necessaria una sperimentazione rigorosa ed una analisi controllata sugli approcci ibridi.

# 3. I modelli valutati e l'approccio proposto.

## 3.1. Il disassemblaggio selettivo multiprodotto.

Nel presente paragrafo ci si sofferma sul sistema di ottimizzazione di una Supply Chain nel campo della logistica inversa, relativamente ai prodotti che ritornano in azienda per essere smantellati, riassemblati e sul recupero dei componenti per la rifabbricazione. In particolare, si analizza un modello di ottimizzazione per un disassemblaggio selettivo multiprodotto, proposto da Gupta.

Lo scopo è quello di fornire un metodo economicamente efficiente con cui le aziende possono chiedere in restituzione vari pezzi di un prodotto per rifabbricarlo.

Il problema è il seguente: **si vuole determinare il numero di prodotti da smantellare in un dato periodo di tempo per soddisfare la richiesta di vari componenti**. Questo richiede un metodo per determinare il tempo ed il numero di prodotti da smantellare per ottenere il numero desiderato di componenti da fornire per la rifabbricazione. Inoltre, se c'è un numero insufficiente di un prodotto disponibile per lo smantellamento, deve essere fatto un ordine di componenti da fonti esterne per soddisfare la richiesta. Nella Figura 3.1 è rappresentata una tecnica per determinare la raccolta di componenti da un mix di prodotti. Questa ricerca si concentra su una serie di decisioni che sono prese per determinare l'insieme di prodotti richiesti che devono essere smantellati per soddisfare la richiesta di numerosi componenti stando attenti al fattore ambientale.

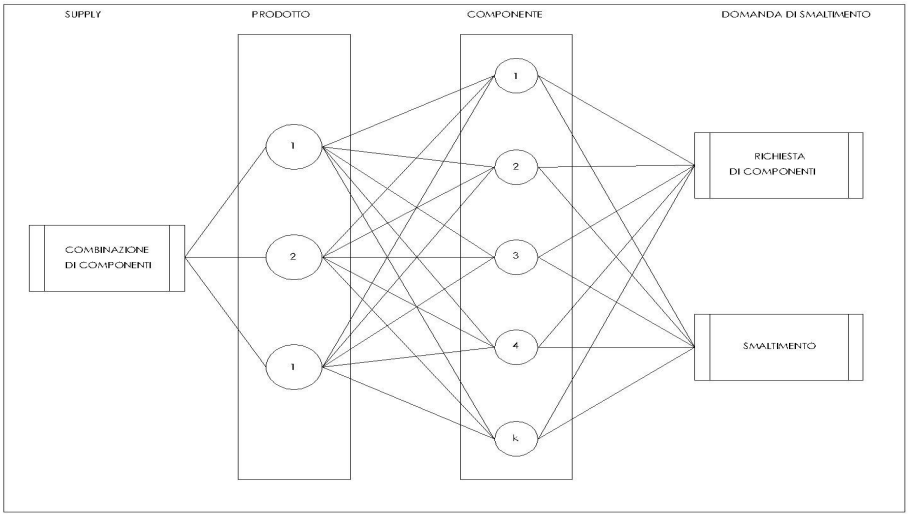



**Figura 3.1– *Richiesta di prodotti/componenti***

"Queste decisioni non possono essere basate tenendo conto solo dei profitti, ma devono anche considerare i costi di smaltimento e distruzione. La formulazione del problema considera una serie di vincoli che devono essere rispettati. Supponiamo che ci siano *n* tipi diversi di prodotti da smantellare per soddisfare la richiesta di varie quantità di *m* componenti ($P_1$,$P_2$,….,$P_n$). La struttura dei componenti, differisce da un prodotto all'altro, questo significa che non tutti i componenti sono in ogni prodotto e potrebbero esserci più componenti della stesso tipo in un unico prodotto. Inoltre, c'e' una serie di vincoli di precedenza da rispettare per smantellare" (Gupta).

Per esempio, una sequenza di disassemblaggio potrebbe essere:

$P_1 \rightarrow P_2 \rightarrow P_3$, oppure un'altra sequenza di disassemblaggio potrebbe essere $P_1 \rightarrow P_3 \rightarrow P_2 \rightarrow P_4$. Per raggiungere il componente $P_3$ nella prima sequenza, è necessario rimuovere prima i componenti $P_1$ e $P_2$, mentre nella seconda, si potrebbe rimuovere solo il componente $P_1$, il risultato è che è economicamente più vantaggioso rimuovere il componente $P_3$ seguendo la seconda sequenza, piuttosto che la prima.

I costi maggiori sono l'acquisizione, il lavoro di smantellamento e smaltimento. Dei tre, i costi di smantellamento sono i più difficile da calcolare.

## 3.2. Il modello

Per arrivare a questi obiettivi, presentiamo una metodologia per determinare il tempo di smantellamento e la raccolta di vari componenti dai prodotti. Al fine di trovare la combinazione di prodotti più economica, è stato sviluppato un modello matematico per la determinazione della richiesta di diversi componenti, il controllo della quantità di prodotti scartati e la ricerca dei costi di smantellamento più bassi.

Il modello afferma che:

1. c'è un ampia gamma di prodotti vecchi;
2. ci sono diversi tipi di prodotti con componenti comuni;
3. la qualità dei componenti è regolare in tutta la linea del prodotto;
4. c'è un periodo dato;
5. è conosciuto il processo di smantellamento;
6. ci sono costi di smantellamento in ogni prodotto lasciato tra quelli utilizzati, ossia che hanno un certo valore.

## 3.3. Soluzione analitica

Nel modello matematico che segue si fa riferimento alla seguente nomenclatura:

| | |
|---|---|
| $\mathbf{A_{ik}}$ | nodo del sottoassieme $k$ nel prodotto $i$ |
| $\mathbf{CP_i}$ | percorso di smantellamento comune dalla lista di componenti nel prodotto $i$ |

| | |
|---|---|
| $\mathbf{D_j}$ | vettore rappresentante la domanda totale dei componenti $P_j$ |
| **DC** | costo di smaltimento dei componenti |
| $\mathbf{DP_i\ (P_j)}$ | percorso del disassemblaggio del componente Pj dal nodo radice del prodotto *i* |
| $\mathbf{DW_j}$ | indice del costo di smantellamento del componente *j* |
| $\mathbf{I_i}$ | vettore riga composto da tutti 1 |
| $\mathbf{I_{ii}}$ | matrice identica di rango *i* |
| $\mathbf{LS(A_{ik})}$ | l'insieme dei componenti e delle parti raggiungibili a partire da $A_{ik}$ |
| $\mathbf{LS^S(A_{ik})}$ | nel disassemblaggio selettivo, l'insieme dei componenti e delle parti raggiungibili a partire da $A_{ik}$ |
| $\mathbf{LS\ (Root_i)}$ | l'insieme dei componenti e delle parti direttamente connesse alla radice *i* |
| $\mathbf{LS^S(Root_i)}$ | nel disessemblaggio selettivo, l'insieme dei componenti e delle parti direttamente connesse alla radice *i* |
| **m** | numero totale di componenti all'interno del problema |
| **MS** | tempo del procedimento di disassemblaggio e di recupero dei componenti dal prodotto |
| **n** | numero totale dei prodotti all'interno del problema |
| $\mathbf{P_j}$ | componente *j* |
| **PC** | costo del processo per unità di tempo. |
| $\mathbf{Q_{ij}}$ | matrice di molteplicità rappresentante il numero di ogni tipo di componente *Pj* ottenuto da ogni tipo di prodotto *i* |
| $\mathbf{Root_i}$ | nodo radice del prodotto *i* |
| $\mathbf{RV_j}$ | Valore di rivendita del componente *j* |
| $\mathbf{s_i}$ | numero totale di nodi subassemblati nel prodotto *i* |
| $\mathbf{S_i}$ | vettore rappresentante la fornitura del prodotto *i* da ogni fonte |
| $\mathbf{Sub_{ik}}$ | nodo di subassemblaggio *k* nel prodotto *i* |
| $\mathbf{T(Root_i)}$ | tempo di disassemblaggio del nodo radice del prodotto *i* in unità di tempo |

| | |
|---|---|
| **$T(A_{ik})$** | tempo di disassemblaggio del subassemblaggio *k* dal prodotto *i* in unità di tempo |
| **$TC_i$** | costo di acquisizione e trasporto del prodotto *i* in dollari per unità |
| **$TD_i$** | tempo totale di disassemblaggio per ogni componente del prodotto *i* per unità di tempo |
| **$TD_i^S$** | tempo totale di disassemblaggio per un insieme di componenti selezionati nel prodotto *i* |
| **TDC** | costo totale dello smaltimento |
| **TPC** | costo totale del processo |
| **TRR** | ricavo totale delle vendite |
| **$W_{ij}$** | matrice rappresentante il numero di unità del componente $P_j$ ottenuto dal prodotto *i* che sarà smaltito |
| **$W_j$** | vettore rappresentante il numero totale di unità del componente $P_j$ che sarà smaltito |
| **$X_{ij}$** | matrice rappresentante il numero di unità di componenti $P_j$ recuperati dal prodotto *i* usati per soddisfare la domanda totale dei componenti |
| **$Y_i$** | vettore rappresentante il numero di ogni prodotto *i* appartenente al lotto da disassemblare |
| **$Y_{ij}$** | matrice rappresentante l'insieme del numero totale dei componenti $P_j$ ottenuti dal prodotto i |
| **Z** | valore da ottimizzare nella funzione obiettivo |

### 3.3.1. Disassemblaggio

Il disassemblaggio inizia con il rimuovere il primo componente da un prodotto. I passi seguenti, disassemblano componenti successive, del prodotto in questione.

***Tempo di disassemblaggio***

*"Il **tempo di disassemblaggio** viene definito come il tempo necessario per disassemblare e recuperare un singolo componente o più componenti" (Gupta).*

Il tempo per disassemblare i componenti di un prodotto è generalmente una misura della prestazione. Il tempo di processo è il fattore dominante in quanto influenza maggiormente i costi del disassemblaggio. Di seguito forniremo un metodo per la determinazione del tempo di disassemblaggio di un prodotto.

Come si evince dalla definizione, un processo di disassemblaggio non implica necessariamente che un prodotto debba essere smontato in ogni sua piccola parte. Invece, il processo di disassemblaggio comprende lo smontaggio di ogni possibile insieme di componenti, che potrebbe, comprendere al suo interno, un unico componente o tutti i componenti presenti nel prodotto.

Tecnicamente la struttura di un prodotto parte da un nodo principale, che è poi seguito da una serie di nodi. Per ottenere un componente gli operatori devono disassemblare il nodo principale ed il nodo, o i nodi successivi, fino a raggiungere il componente richiesto. Se da un nodo principale si ottengono più componenti, il tempo di disassemblaggio di un nodo principale deve essere calcolato una sola volta e non per ogni componente ottenuto dal nodo, lo stesso discorso, ovviamente, vale non solo per il nodo principale, ma anche per gli altri nodi secondari. Ciò significa che il tempo di disassemblaggio di un nodo non dipende da quanti componenti si ricavano da esso.

*"Si definisce **percorso di disassemblaggio** (Disassembly path DP) di un componente (dal nodo principale allo stesso componente) un insieme ordinato di nodi di disassemblaggio che portano fino al nodo secondario in questione".*

*Si definisce **percorso comune di disassemblaggio** (common disassembly path CP) di due (o più) componenti, l'insieme ordinato di nodi di disassemblaggio che sono uguali nei due percorsi di disassemblaggio dei componenti in questione.*

Si considerino due prodotti G e H ognuno costituito da 4 componenti $P_1$, $P_2$, $P_3$ e $P_4$, come mostrato in Figura 3.2. Nel prodotto G il percorso del disassemblaggio ($DP_g$) per i componenti $P_3$ e $P_4$ sono rispettivamente:

$$DP_G(P_3) = \{Root_G, Sub_{G,B}\}$$

$$DP_G(P_4) = \{Root_G, Sub_{G,B}\}$$

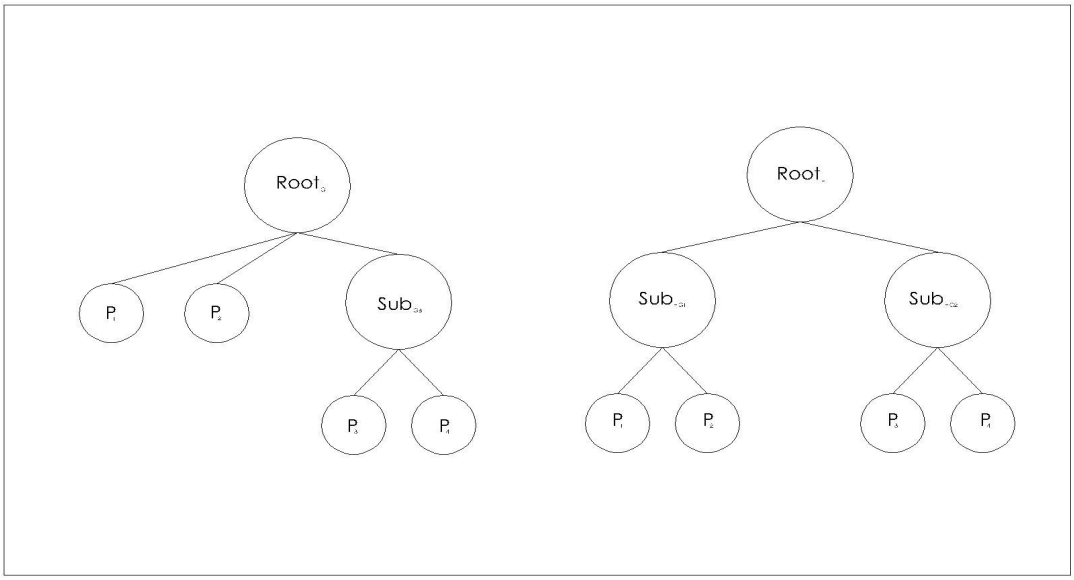


**Figura 3.2: Strutture del prodotto G e del prodotto H**

Così il percorso di disassemblaggio per entrambi i componenti è comune, ed è rappresentato dalla seguente formula:

$$CP_G(P_3, P_4) = \{Root_G, Sub_{G,b}\}$$

$P_3$ e $P_4$ sono accessibili solo dopo aver smontato $Root_G$ e $Sub_{G,b}$. Se dovessimo scegliere $P_1$ e $P_3$ invece di $P_3$ e $P_4$, l'unico nodo in comune nei due percorsi sarebbe $Root_G$, (che ovviamente è in comune a qualsiasi percorso possibile in questo esempio).

*Per ogni nodo di subassemblaggio vi è un insieme di nodi secondari che lo succedono (leaf successor set LS). In questo modo l'LS di un nodo principale, comprende tutti i nodi secondari della struttura del prodotto, mentre per ogni nodo secondario vi è un insieme differente che gli succede e questi è diverso dagli insiemi che derivano dagli altri nodi della struttura.*

Il tempo totale di disassemblaggio *completo* per un prodotto può essere calcolato come segue:

$$TD_i = \left( \underset{\forall P_j \in LS(Root_i)}{Max} \left[ \frac{\{D_j\}}{\{Q_{ij}\}} \right] \right) (T(Root_i)) + \sum_{k=1}^{Si} \left\{ \left( \underset{\forall P_j \in LS(A_{jk})}{Max} \left[ \frac{\{D_j\}}{\{Q_{ij}\}} \right] \right) (T(A_{ik})) \right\} \quad \text{(e 3.1)}$$

In tale formula, il primo membro della somma rappresenta il tempo del disassemblaggio del root, mentre il secondo costituisce la sommatoria di tutti i sub del prodotto.

Allo stesso modo, il tempo totale, per un disassemblaggio *selettivo*, ossia mirato solo ad alcuni determinati prodotti, può essere calcolato come segue:

$$TD^s{}_i = \left( \underset{\forall P_j \in LS^s(Root_i)}{Max} \left[ \frac{\{D_j\}}{\{Q_{ij}\}} \right] \right) (T(Root_i)) + \sum_{K=1}^{Si} \left\{ \left( \underset{\forall P_j \in LS^s(A_{jk})}{Max} \left[ \frac{\{D_j\}}{\{Q_{ij}\}} \right] \right) (T(A_{ik})) \right\} \quad \textbf{(e 3.2)}$$

### *3.3.2. Metodologia per determinare l'insieme dei componenti*

Consideriamo un insieme di prodotti *i*, dove $1 \le i \le n$ e consideriamo $Y_i$, come il vettore rappresentante il numero di ogni tipo di prodotto *i*

nell'insieme da disassemblare. Ancora, consideriamo $Q_{ij}$, quale matrice rappresentante il numero di ogni tipo di componente, $P_j$, con $1 \le j \le m$, contenuto in ogni tipo di prodotto i, se $I_{ii}$ rappresenta una matrice identica (i x i), allora $Y_{ij}$, rendimento delle unità dei componenti ottenuti dai prodotti *i*, può essere ottenuto come segue:

$$Y_{ij} = (Y_i \cdot I_{ii}) \cdot Q_{ij} \qquad \textbf{(e 3.3)}$$

Supponiamo che i componenti ricavati dal disassemblaggio di determinati prodotti, o soddisfano la domanda, o vengono smaltiti immediatamente. Se rappresentiamo con le matrici $X_{ij}$ e $W_{ij}$ rispettivamente il numero delle unità dei componenti ottenuti dal prodotto i che soddisfano la domanda e che vengono smaltiti, si avrà:

$$Y_{ij} = X_{ij} + W_{ij} \qquad \textbf{(e 3.4)}$$

Se, ancora, definiamo con $D_j$ e $W_j$, due vettori rappresentanti il numero delle unità dei componenti usati per soddisfare la domanda, il primo, e quelli destinati allo smaltimento, il secondo, ed ancora con $I_i$ il vettore riga di tutti 1, avremo:

$$D_j = I_i \cdot X_{ij} \qquad \text{(e 3.5)}$$

e

$$W_j = I_i \cdot W_{ij} \qquad \text{(e 3.6)}$$

## 3.4. Modello di ottimizzazione

A questo punto, presentiamo un modello di ottimizzazione, per determinare un insieme di prodotti da smantellare per soddisfare la richiesta di vari componenti. Attraverso la programmazione lineare, la formulazione del problema avviene come segue:

"La funzione obiettivo consiste di tre termini principali; essi sono: *ricavo totale delle vendite* (TRR), *costo totale del processo* (TPC), *costo di smaltimento totale* (TDC)".

- RICAVO TOTALE DI RIVENDITA: $RV_j$ e $TC_i$, influenzano direttamente il TRR. RV è il valore di rivendita del componente j, e TCi è il costo di acquisizione e trasporto del prodotto i (per unità di tempo). L' equazione di ricavo, può essere formulata come il valore del ricavo, meno il costo totale dell'acquisizione del prodotto:

$$TRR = \sum_i \sum_{\substack{j \exists D_i > 0 \\ P_j \in LS^s(Root)}} \left(RV_j \cdot \{X_{ij}\}\right) - \sum_i \left(TC_i \cdot \{Y_i\}\right) \qquad \textbf{(e 3.7)}$$

- COSTO DEL PROCESSO TOTALE: può essere formulato moltiplicando il tempo per il disassemblaggio ed il recupero dei componenti dai prodotti (MS) e il costo totale del processo per unità di tempo (PC):

$$TPC = PC \cdot MS \qquad \textbf{(e 3.8)}$$

Dove MS può essere calcolato da :

$$MS = \sum_i TD_i^S \qquad \textbf{(e 3.9)}$$

e, di conseguenza, $TD_i^S$, può essere calcolato, come visto precedentemente

$$TD^s{}_i = \left( \frac{Max}{\forall P_j \in LS^s(Root_i)} \left[ \frac{\{D_j\}}{\{Q_{ij}\}} \right] \right) (T(Root_i)) + \sum_{K=1}^{Si} \left\{ \left( \frac{Max}{\forall P_j \in LS^s(A_{jk})} \left[ \frac{\{D_j\}}{\{Q_{ij}\}} \right] \right) (T(A_{ik})) \right\} \quad \text{(e 3.2)}$$

come segue:

$$TD^s{}_i = \left( \frac{Max}{\forall P_j \in LS^s(Root_i)} \left[ \frac{\{X_{ij}\}}{\{Q_{ij}\}} \right] \right) (T(Root_i)) + \sum_{K=1}^{Si} \left\{ \left( \frac{Max}{\forall P_j \in LS^s(A_{jk})} \left[ \frac{\{X_{ij}\}}{\{Q_{ij}\}} \right] \right) (T(A_{ik})) \right\} \quad \textbf{(e 3.10)}$$

- COSTO TOTALE DI SMALTIMENTO è calcolato moltiplicando il totale di smaltimento per il numero dei componenti da smaltire, nel modo seguente:

$$TDC = DC \cdot \left( \sum_i \sum_{\substack{j \exists D_j > 0 \\ P_j \in LS^s(Root_i)}} (DW_j \cdot \{W_{ij}\}) \right) + DC \cdot \left( \sum_i \sum_{\substack{j \exists D_j > 0 \\ P_j \in LS^s(Root_i)}} (DW_j \cdot \{(Y_i \cdot I_{ii}) \cdot Q_{ij}\}) \right) \quad \textbf{(e 3.11)}$$

In tale formulazione, la prima parte è relativa allo smaltimento dei prodotti che hanno valore per un riutilizzo diverso da zero ($RV_j \neq 0$), mentre la seconda è legata ai componenti rimanenti.

Da notare che $DW_j$ è l'indice del costo di smaltimento, che può variare da 1 a 10, in funzione della difficoltà dello smaltimento del componente j. Wij può essere scritta dall'equazione $Y_{ij} = X_{ij} + W_{ij}$

(e 3.4) come segue:

$$W_{ij} = Y_{ij} - X_{ij} \quad \textbf{(e 3.12)}$$

Se la domanda dei componenti j è 0, allora $X_{ij}$ sarà 0 e $W_{ij}$ può essere scritta come segue:

$$W_{ij} = (Y_{ij} \cdot I_{ii}) \cdot Q_{ij} \quad \textbf{(e 3.13)}$$

L'obiettivo del modello è massimizzare il ricavo dovuto al recupero, quindi, la funzione obiettivo, si scrive come segue:

$$Maximize Z = TRR - TPC - TDC \qquad \text{(e 3.14)}$$

A questo punto possiamo definire i vincoli del nostro problema:

- **Vincoli fornitura/Domanda**.

  Il numero dei prodotti nel lotto da disassemblare, per soddisfare la domanda dei componenti, non deve eccedere il numero dei prodotti disponibili. Così:

$$\{Y_i\} \leq \{S_i\} \qquad \text{(e 3.15)}$$

- **Vincoli della struttura del prodotto**.

  Dalle equazioni $Y_{ij} = (Y_i \cdot I_{ii}) \cdot Q_{ij}$ (e 3.3) e $Y_{ij} = X_{ij} + W_{ij}$ (e 3.4), possiamo ricavare la seguente espressione:

$$\{X_{ij}\} + \{W_{ij}\} = \{(Y_i \cdot I_{ii}) \cdot Q_{ij}\} \qquad \textbf{(e 3.16)}$$

Per ogni i ed ogni j $\exists D_j > 0$ e

$$P_j \in LS^s(Root_i)$$

- **Vincoli per il soddisfacimento della domanda dei componenti**

  Dall'equazione 5 si ricava:

$$\{I_i \cdot X_{ij}\} = \{D_j\} \qquad \text{(e 3.17)}$$

per ogni

$$j \exists D_j > 0 \quad \text{e} \quad P_j \in LS^s(Root_i)$$

- **Vincoli di numeri interi e non negativi.**

La fornitura dei prodotti e la domanda dei componenti deve essere di valori positivi ed interi.

$$\{Y_i\}, \{X_{ij}\} \geq 0 \qquad \text{(e 3.18)}$$

## 3.5. Esame di un caso

Consideriamo un caso specifico, per mostrare praticamente l'applicazione del modello di ottimizzazione di disassemblaggio dei componenti.

Supponiamo di avere tre modelli di computer, Proliant 6000 (Figura 3.3), 6500 (Figura 3.4) e 7000 (Figura 3.5), da smantellare. L'obiettivo è trovare un insieme ottimale dei tre modelli da smantellare per far fronte ad una richiesta selettiva di componenti, in modo tale da massimizzare l'entrata dovuta al recupero di tali componenti.

Le tabelle e le figure, rappresentano i dati di cui abbiamo bisogno per il nostro caso.

Si consideri che i tre differenti prodotti, sono varie combinazione di un massimo di 27 diversi componenti. La Tabella 3.1, mostra i dati relativi alla domanda, il valore, e l'indice del costo di smantellamento per ogni componente. In Tabella 3.2, vengono mostrati i tempi necessari per disassemblare i nodi principali e secondari per ogni prodotto.

Ulteriori dati sono:

$$S_i = [500,500,500]$$
$$PC = 0.2$$
$$DC = 0.25$$
$$TC_1 = 140$$
$$TC_2 = 120$$
$$TC_3 = 135.$$

Il caso in questione, può quindi essere formulato come segue:

$$\text{Maximize} Z = \text{TRR} - \text{TPC} - \text{TDC}$$
$$\{Y_i\} \le \{S_i\}$$
$$\{X_{ij}\} + \{W_{ij}\} = \{(Y_i \cdot I_{ii}) \cdot Q_{ij}\}_j \exists D_j > 0$$
$$P_j \in LS^s(Root_i)$$
$$\{I_i \cdot X_{ij}\} = \{D_i\}_j \exists D_j > 0 \;\; P_j \in LS^s(Root_i)$$
$$\{Y_i\}\{X_{ij}\} \ge 0 \;_j \exists D_j > 0 \; 1 \le i \le n; 1 \le j \le m$$
$$\text{TRR} = \sum_i \sum_{\substack{j \exists D_j > 0 \\ P_j \in LS^s(Root_i)}} (RV_i \cdot \{X_{ij}\}) - \sum_i (TC_i \cdot \{Y_i\})$$
$$\text{TPC} = PC \cdot \sum_i TD_i^s$$
$$\text{TDC} = DC \cdot \left( \sum_i \sum_{\substack{j \exists D_j > 0 \\ P_j \in LS^s(Root_i)}} (DW_j \cdot \{W_{ij}\}) \right) + DC \cdot \left( \sum_i \sum_{\substack{j \exists D_j > 0 \\ P_j \in LS^s(Root_i)}} (DW_j \cdot \{(Y_i \cdot I_{ii}) \cdot Q_{ij}\}) \right)$$
$$TD^s{}_i = \left( \frac{\text{Max}}{\forall P_j \in LS^s(Root_i)} \left[ \frac{\{X_{ij}\}}{\{Q_{ij}\}} \right] \right) (T(Root_i)) +$$
$$+ \sum_{k=1}^{Si} \left\{ \left( \frac{\text{Max}}{\forall P_j \in LS^s(A_{jk})} \left[ \frac{\{X_{ij}\}}{\{Q_{ij}\}} \right] \right) (T(A_{ik})) \right\}$$

Ritornando alla formula iniziale Z =TRR-TPC-TDC otteniamo che il valore ottimale della funzione obiettivo è € 4.907,90 con un valore del TRR pari a 14.055,00 con un valore del TPC pari a 1.272,60 e un valore di TDC pari a 7.874,50. I valori di Y1, Y2, e Y3 calcolati dal software sono 237, 200 e 163, relativi, rispettivamente, alle tre tipologie di computer Proliant server models 6000, 6500 e 7000. Nella Tabella 3.1 e Tabella 3.2 seguenti, sono riportati i dati, nella Figura 3.3, , Figura 3.4 e Figura 3.5 sono descritte, graficamente, le strutture dei tre tipi diversi di prodotti con la rappresentazione, per ciascuno di essi, della presenza dei root, sub e componenti e nella Tabella 3.1, invece, sono riportati i risultati di tale elaborazione:

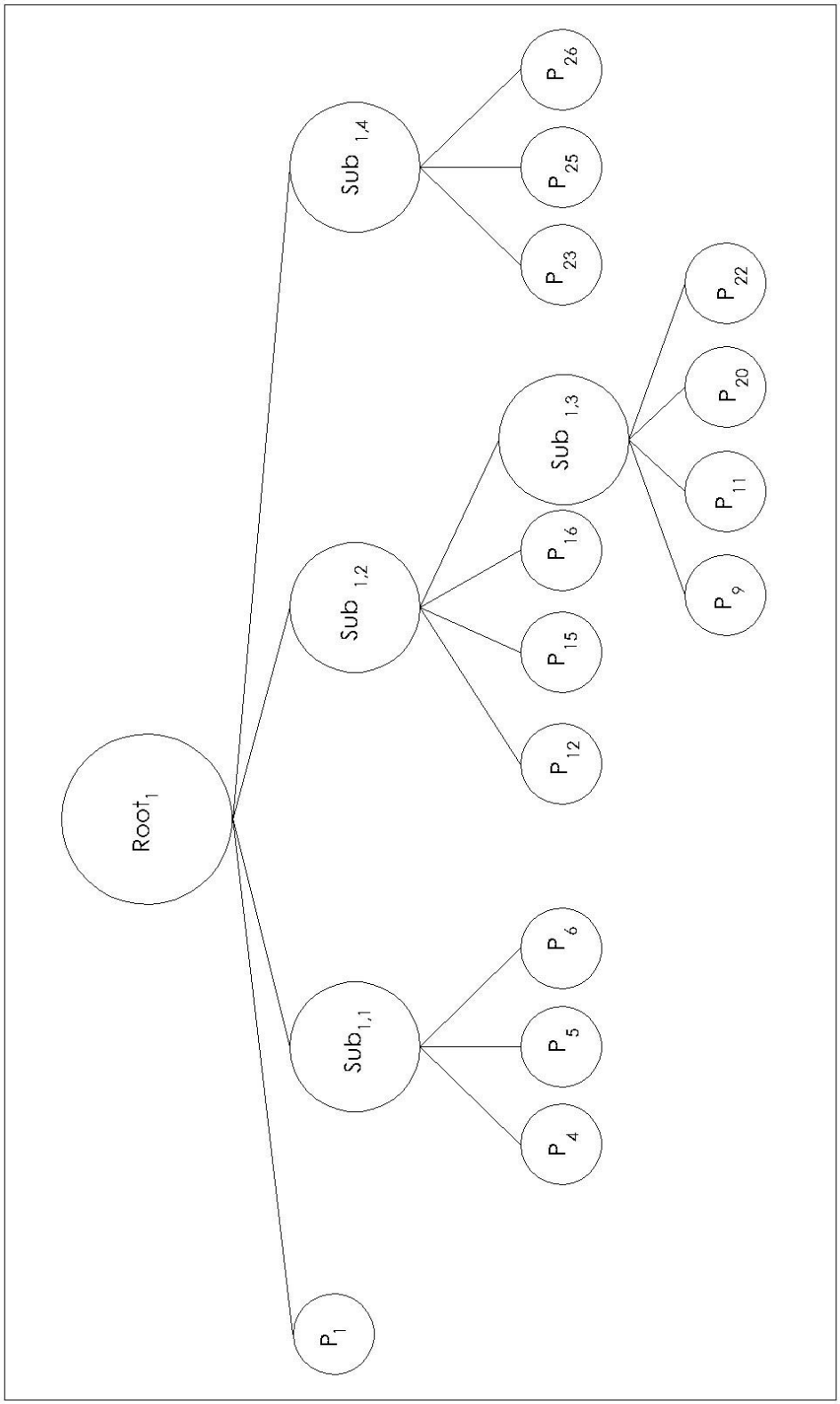



**Figura 3.3– Struttura Proliant server models 6000**

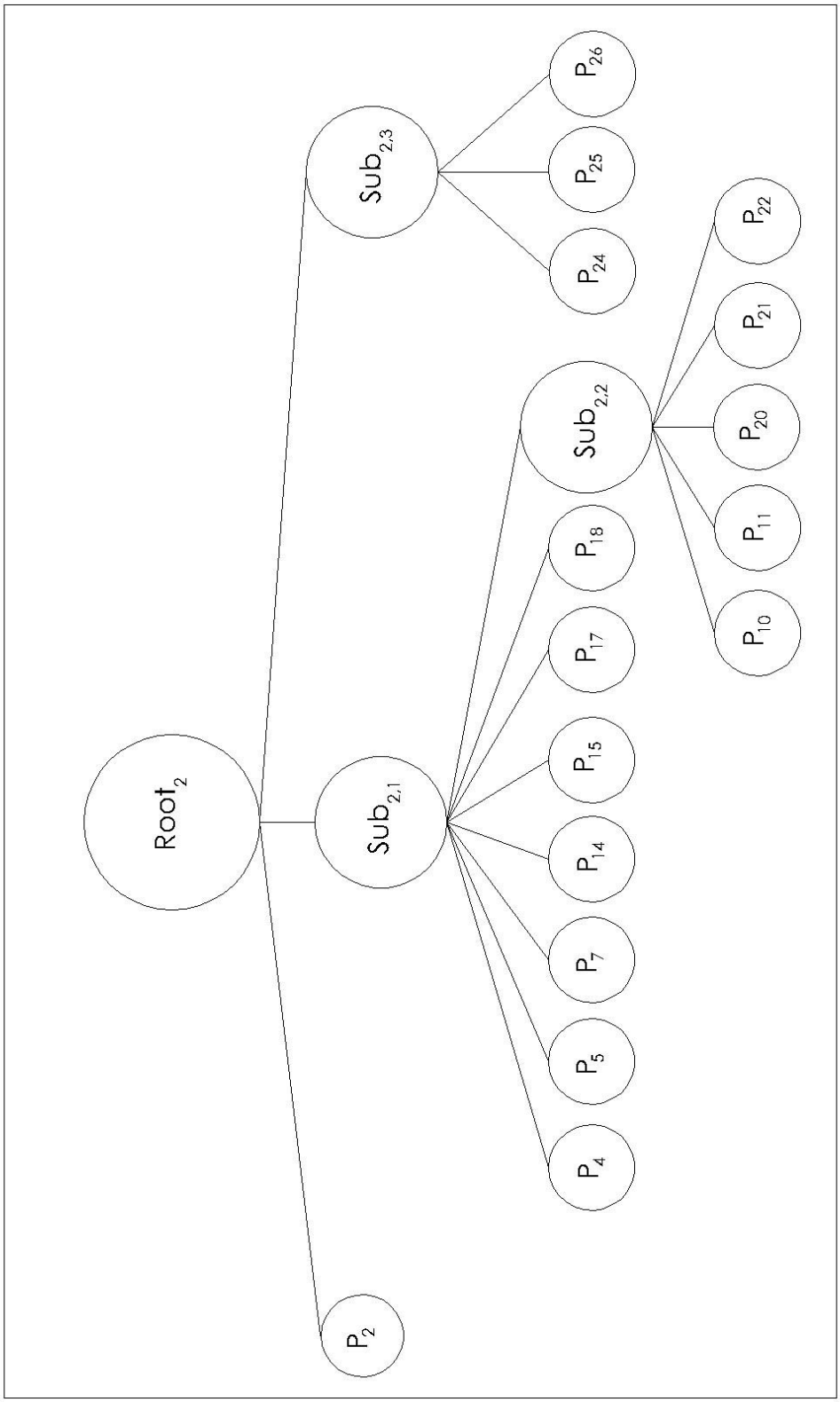



**Figura 3.4- Struttura Proliant server models 6500**

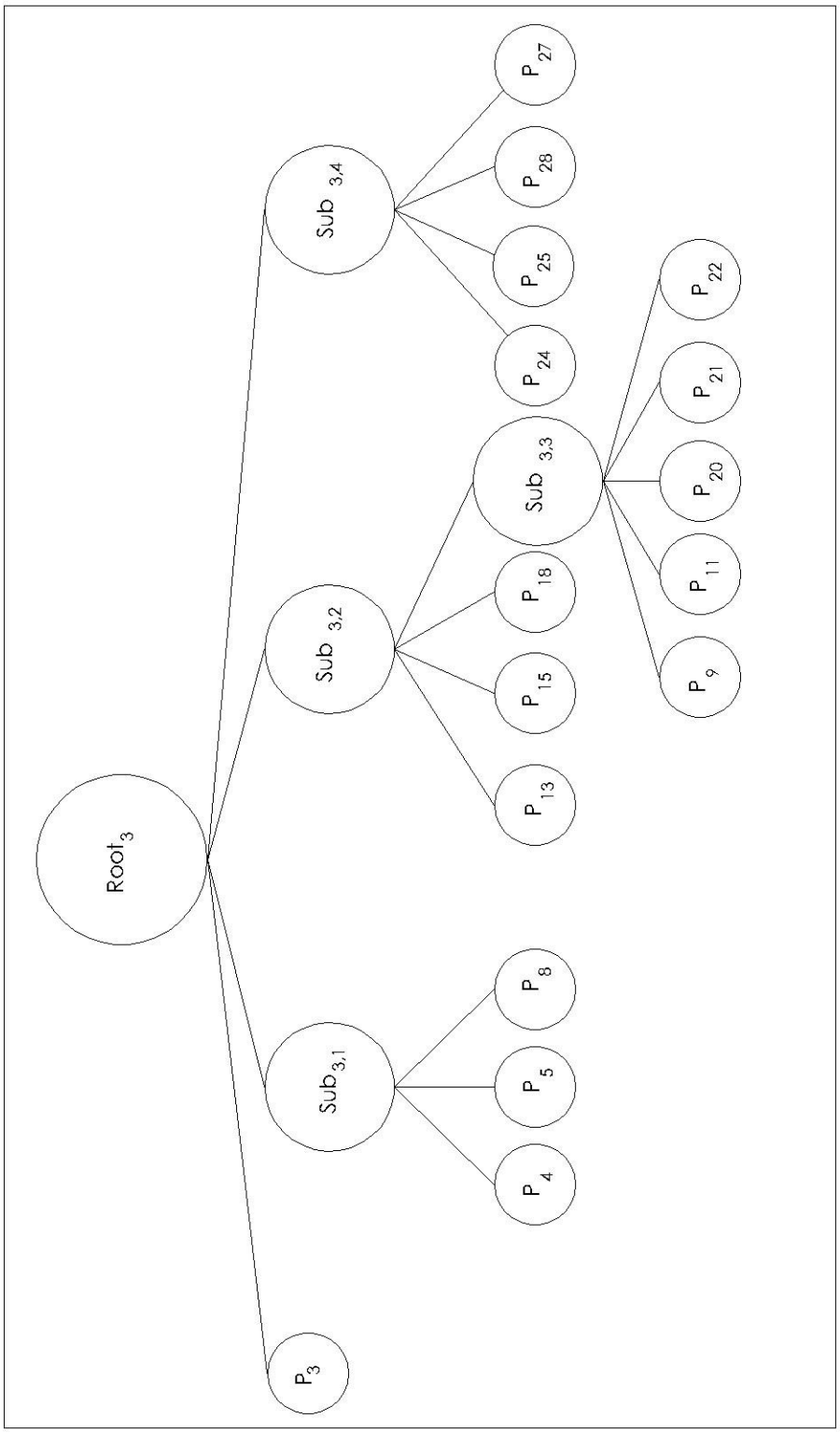



**Figura 3.5- Struttura Proliant server models 7000**

| Numero dei componenti | Nome dei componenti | Numero seriale | Molteplicità/Comunanza (Qlj) | | | Domanda (Dj) | Valore (RVj) | Indice del costo di smaltimento (DWj) |
|---|---|---|---|---|---|---|---|---|
| | | | L6000 | L6500 | L7000 | (unit) | (S/Unit) | (1=lowest,10=highest) |
| 1 | Housing Assembly (L6000) | 186893-001 | 1 | - | - | - | - | 5 |
| 2 | Housing Assembly (L 6500) | 169287-001 | - | 1 | - | - | - | 9 |
| 3 | Housing Assembly (L 7000) | 306367-001 | - | - | 1 | - | - | 6 |
| 4 | Integraded Management Display With cable | 271930-001 | 1 | 1 | 1 | - | - | 2 |
| 5 | Power Supply | 169286-001 | 1 | 2 | 2 | 550 | 2 | 10 |
| 6 | Twin Fans with Bracket (L6000) | 289743-001 | 1 | - | - | - | - | 3 |
| 7 | Hot- Plug Fan Assembly (L 6500) | 241708-001 | - | 1 | - | - | - | 3 |
| 8 | Hot- Plug Fan Assembly (L 7000) | 300362-001 | - | - | 1 | - | - | 4 |
| 9 | 686 Processor Board (L 6000/ 7000) | 186889-001 | 1 | - | 1 | 400 | 15 | 5 |
| 10 | 686 Processor Board (L 6500) | 169291-001 | - | 2 | - | 120 | 18 | 5 |
| 11 | 686/200 MHz Processor And Heat Sink | 296492-001 | 2 | 4 | 4 | 1150 | 18 | 2 |
| 12 | PCI/ EISA Expansion Board ( L 6000) | 186888-001 | 1 | - | - | - | - | 1 |
| 13 | PCI/ EISA Expansion Board ( L7000) | 296279-001 | - | - | 1 | - | - | 1 |
| 14 | I/O Board (L 6500) | 169486-001 | - | 1 | - | - | - | 1 |
| 15 | SCSI Adapter | 189638-001 | 1 | 1 | 1 | - | - | 1 |
| 16 | PCI Board 10/100 NIC (L 6000) | 169849-001 | 1 | - | - | - | - | 3 |
| 17 | PCI Board 10/100 NIC (L 6500) | 242560-001 | - | 1 | - | - | - | 3 |
| 18 | PCI Board 10/100 NIC (L 7000) | 242560-001 | - | - | 1 | - | - | 3 |
| 19 | Fan Control Board (L 6500) | 169289-001 | - | 1 | - | 200 | 14 | 2 |
| 20 | Memory module, 64 MB, 60ns, EDO | 281858-001 | 6 | 4 | 4 | 1250 | 15 | 1 |
| 21 | Memory module, 128 MB, 60ns, EDO (L 6500/7000) | 281859-001 | - | 2 | 4 | 1050 | 25 | 1 |
| 22 | Memory Expansion Board with Stiffener | 289745-001 | 1 | 1 | 1 | - | - | 5 |
| 23 | 1,44 MB Diskette Drive (L 6000) | 296224-001 | 1 | - | - | - | - | 6 |
| 24 | 1,44 MB Diskette Drive (L 6500/7000) | 144207-001 | - | 1 | 1 | - | - | 6 |
| 25 | 16X CD- ROM Drive | 278791-001 | 1 | 1 | 1 | 580 | 6 | 5 |
| 26 | 9.1 GB Hot-Pluggable SCSI Hard Drive | 199888-001 | 1 | 2 | 2 | 450 | 15 | 7 |
| 27 | 4 GB Hot-Pluggable SCSI Hard Drive (L7000) | 242622-001 | - | - | 7 | 350 | 15 | 3 |

**Tabella 3.1– DatiEsempio**

| DISASSEMBLY TIME | | | | | |
|---|---|---|---|---|---|
| L6000 | | L8500 | | L7000 | |
| Subassembly | Time | Subassembly | Time | Subassembly | Time |
| T (Root $_1$) | 3.7 | T (Root $_2$) | 4.5 | T (Root $_3$) | 3.7 |
| T (Sub $_{1,1}$) | 2.5 | T (Sub $_{2,1}$) | 3.1 | T (Sub $_{3,1}$) | 2.6 |
| T (Sub $_{1,2}$) | 1.9 | T (Sub $_{2,2}$) | 2.3 | T (Sub $_{3,2}$) | 2.0 |
| T (Sub $_{1,3}$) | 2.1 | T (Sub $_{2,3}$) | 2.3 | T (Sub $_{3,3}$) | 2.1 |
| T (Sub $_{1,4}$) | 1.7 | | | T (Sub $_{3,4}$) | 1.7 |

**Tabella 3.2– Dati esempio**

| NUMERO | | DOMANDA | | | TOTALE | Smaltimento dei componenti (WIJ) | | | TOTALE |
|---|---|---|---|---|---|---|---|---|---|
| NUMERO DI SERIE | | L6000 | L6500 | L7000 | | L6000 | L6500 | L7000 | |
| 1 | 186893-001 | - | - | - | - | 237 | - | - | 237 |
| 2 | 169287-001 | - | - | - | - | - | 200 | - | 200 |
| 3 | 306367-001 | - | - | - | - | - | - | 163 | 163 |
| 4 | 271930-001 | - | - | - | - | 237 | 200 | 163 | 600 |
| 5 | 169286-001 | 0 | 400 | 150 | 550 | 237 | 0 | 176 | 413 |
| 6 | 289743-001 | - | - | - | - | 237 | - | - | 237 |
| 7 | 241708-001 | - | - | - | - | - | 200 | - | 200 |
| 8 | 300362-001 | - | - | - | - | - | - | 163 | 163 |
| 9 | 186889-001 | 237 | - | 163 | 400 | 0 | - | 0 | 0 |
| 10 | 169291-001 | - | 120 | - | 120 | - | 280 | - | 280 |
| 11 | 296492-001 | 0 | 498 | 652 | 1150 | 474 | 302 | 0 | 776 |
| 12 | 186888-001 | - | - | - | - | 237 | - | - | 237 |
| 13 | 296279-001 | - | - | - | - | - | - | 163 | 163 |
| 14 | 169486-001 | - | - | - | - | - | 200 | - | 200 |
| 15 | 189638-001 | - | - | - | - | 237 | 200 | 163 | 600 |
| 16 | 169849-001 | - | - | - | - | 237 | - | 163 | 400 |
| 17 | 242560-001 | - | - | - | - | - | 200 | - | 200 |
| 18 | 242560-001 | - | - | - | - | - | - | 163 | 163 |
| 19 | 169289-001 | - | 200 | - | 200 | - | 0 | - | 0 |
| 20 | 281858-001 | 21 | 800 | 429 | 1250 | 1401 | 0 | 223 | 1624 |
| 21 | 281859-001 | - | 398 | 652 | 1050 | - | 2 | 0 | 2 |
| 22 | 289745-001 | - | - | - | - | 237 | 200 | 163 | 600 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 23 | 296224-001 | - | - | - | - | 237 | - | - | 237 |
| 24 | 144207-001 | - | - | - | - | - | 200 | 163 | 363 |
| 25 | 278791-001 | 237 | 200 | 143 | 580 | 0 | 0 | 20 | 20 |
| 26 | 199888-001 | 0 | 124 | 326 | 450 | 237 | 276 | 0 | 513 |
| 27 | 242622-001 | - | - | 350 | 350 | | - | 791 | 791 |

**Tabella 3.3- Risultati esempio**

## 3.6. Algoritmo di dimensionamento del lotto

I problemi dinamici di dimensionamento del lotto con vincoli di lot-size, hanno come funzione obiettivo la minimizzazione della variabile dei costi di produzione, dopo una programmazione finita di intervalli temporali. Tale variabile considera compresi anche i costi di immagazzinamento e quindi delle scorte ed i costi di setup. La pianificazione del periodo di riferimento è ottenuta tramite la suddivisione in diversi periodi *t* e tale orizzonte temporale è rappresentato dalla durata *T*. Per ogni periodo della pianificazione temporale, la domanda finale di ogni item è assunta nota ed è considerata eseguita senza ritardo. I costi di magazzino sono calcolati in base alle scorte alla fine del periodo. I costi ed i tempi di setup provengono, per un item, da ogni periodo di produzione.
Le risorse hanno una capacità limitata per periodo, che però può essere aumentata facendo ricorso allo straordinario. Il modello così formulato, anche chiamato CLSPL (Capacitated Lot-Sizing Problem whit Linked Lot-Sizes) è:

$$\text{Min!} \quad \Sigma_{j=1..J} \Sigma_{t=1..T} [h_j * I_{jt}] + \Sigma_{j=1..J} \Sigma_{t=1..T} [sc_j * (Y_{jt} - W_{jt})] + \Sigma_{m=1..M} \Sigma_{t=1..T} [oc_{mt} * O_{mt}] \qquad \text{(e 3.19)}$$

Soggetto a:

$$I_{jt-1} + X_{jt} = P_{jt} + \Sigma_{k \in Sj} (r^d_{jk} * X_{kt}) + I_{jt} \qquad \forall j = 1,...,J; t = 1,...,T \qquad \text{(e 3.20)}$$

$$\Sigma_{j \in Rm} (a_{mj} * X_{jt}) + \Sigma_{j \in Rm} [st_{jm} * (Y_{jt} - W_{jt})] \le C_{mt} + O_{mt} \qquad \forall m = 1,...M; t = 1,...,T \qquad \text{(e 3.21)}$$

$$X_{jt} \le B_{jt} * Y_{jt} \qquad \forall j = 1,...J; t = 1,...,T \qquad \text{(e 3.22)}$$

$$\Sigma_{j \in Rm} W_{jt} \le 1 \qquad \forall m = 1,...,M; t = 2,...,T \qquad \text{(e 3.23)}$$

$$W_{jt} \le Y_{j\,t-1} \qquad \forall j = 1,...,J; t = 2,...,T \qquad \text{(e 3.24)}$$

$$W_{jt} \le Y_{j\,t} \qquad \forall j = 1,...,J; t = 2,...,T \qquad \text{(e 3.25)}$$

$1 - \Sigma_{j \in Rm} Y_{jt} + J*Q_{mt} \geq 0 \qquad \forall m = 1,...,M; t = 2,...,T-1$ (e 3.26)

$W_{jt+1} + W_{jt} + Q_{mt} \leq 2 \qquad \forall m = 1,...,M; j \in R_m; t = 2,...,T-1$ (e 3.27)

$0 \leq O_{mt} \leq 2 \quad \forall j = 1,...,J; t = 1,...,T$ (e 3.28)

$X_{jt} \geq 0 \; \forall m = 1,...,M; t = 1,...,T$ (e 3.29)

$0 \leq Q_{mt} \leq 1 \qquad \forall m = 1,...,M; t = 2,...,T-1$

$I_{jt} \geq 0 \quad \forall j = 1,...,J; t = 1,...,T$

$Y_{jt} \in \{0;1\} \quad \forall j = 1,...,J; t = 1,...,T$

$W_{jt} \in \{0;1\} \quad \forall j = 1,...,J; t = 2,...,T$

Dove gli indici rappresentano:

*j* : items o operazioni ( prodotti finiti, prodotti intermedi, materie prime), *j=1,…,J*
*m* : Resources ( personale, macchine, linee produttive), *m=1,…,M*
*t* : Periodi, *t=1,…,T*
$R_m$ : Set di prodotti realizzabili sulla risorsa *m*
$S_j$ : Set dei componenti immediatamente successivi all'item *j* nella distinta base

Dati:

$a_{mj}$ : Capacità necessaria su di una risorsa *m* per una unità dell'item *j*
$B_{jt}$ : Limite superiore della dimensione del lotto dell'item *j* nel periodo *t*
$C_{mt}$ : Capacità disponibile della risorsa *m* nel periodo *t*
$h_j$ : Costi di magazzino per una unità dell'item *j* per un periodo
$oc_{mt}$ : Costi di straordinario per una unità della risorsa *m* nel periodo *t*
$P_{jt}$ : Domanda di prodotti finiti per l'item *j* nel periodo *t*
$r^d_{jk}$ : Numero di unità dell'item *j* richieste per produrre una unità dell'item *k*, immediatamente superiore
$sc_j$ : Costi di setup per un lotto di item *j*
$st_{jm}$ : Tempi di setup per l'item *j* sulla risorsa *m*

Variabili:

$I_{jt}$ : Scorte dell'item $j$ alla fine del periodo $t$

$O_{mt}$ : Quantità di straordinario sulla risorsa $m$ usata nel periodo $t$

$Q_{mt}$ : Indicatore della produzione del singolo item (=1, se c'è la produzione di al massimo 1 item sulla risorsa $m$ nel periodo $t$

$W_{jt}$ : Variabile binaria di connessione (=1, , se la produzione dell'item $j$ è distribuita tra il periodo $t$-$1$ e $t$, 0 altrimenti)

$X_{jt}$ : Quantità prodotta dell'item $j$ nel periodo $t$ (lot-size)

$Y_{jt}$ : Variabile binaria di setup (=1, se c'è setup dell'item $j$ nel periodo $t$, 0 altrimenti)

La funzione obiettivo, tende alla minimizzazione della somma dei costi di magazzino, dei costi di setup e dei costi di straordinario. Tutti gli altri costi di produzione sono assunti fissati e indipendenti dal tempo, di conseguenza i costi di produzione diretti non sono attribuiti al lot-size $X_{jt.}$ La domanda all'interno dei periodi degli items può essere divisa tra i lotti prodotti in periodi diversi.

I vincoli di bilanciamento multi-livello (e 3.20) assicurano che non ci saranno ritardi. La produzione multi-livello di un lotto dell'item $k$, produrrà una domanda dipendente degli items $j$ immediatamente successori. La capacità richiesta per la produzione di un lotto non può eccedere la capacità normalmente disponibile (estendibile tramite l'impiego dello straordinario) (e 3.21). La capacità produttiva delle risorse è limitata e rinnovabile. Le capacità richieste si ottengono sia dai tempi di produzione per item per i quantitativi prodotti, sia dai tempi di setup necessari per ogni lotto. Il vincolo di setup (e 3.22) impone alla variabile $Y_{jt}$ di essere pari ad uno, in caso di un lotto dell'item $j$ prodotto in un periodo $t$.

Il vincolo (e 3.23), assicura che al più un setup può essere mantenuto tra due periodi successivi. I vincoli (e 3.24) e (e 3.25) legano le variabili di connessione alle variabili di setup, sebbene i vincoli (e 3.26) e (e 3.27) controllino che i setup siano calcolati correttamente se lo stesso setup è mantenuto tra due periodi di tempo successivi. Il vincolo (e 3.28) impone che le ore di straordinario per ogni macchina e ogni periodo, non siano maggiori di 2. Nelle relazioni vincolari (e 3.29), tutte le variabili sono state ristrette ad assumere valori non negativi o binari.

## 3.7. Integrazioni e modifiche apportate agli algoritmi

L'algoritmo di dimensionamento del lotto appena esaminato, non prevede una funzione di costo legata al ritardo (backlog), la possibilità di far ricorso a terzi per la produzione di alcuni prodotti e la possibilità di utilizzare componenti provenienti da prodotti EOL (end of life). Sono state studiate, quindi, una serie di modifiche da apportare al modello per considerare tali vincoli. Siano:

- $Cr_{jt}$ : costo per il ritardo dell'item j nel periodo t
- $Vrit_{jt}$ : variabile relativa all'item j in ritardo nel periodo t
- $Cext_j$: costo per la produzione esterna dell'item j
- $Vext_{jt}$: produzione relativa all'item j che viene esternalizzata al periodo t
- $U_{jt}$: quantità di prodotto relativa all'item j al tempo t proveniente da prodotti EOL

Pertanto, il costo per il ritardo sarà dato dalla quantità:

$$\sum_{j=1...J} \sum_{t=1...T} [Cr_{jt} * Vrit_{jt}]$$

il costo per la produzione commissionata esternamente all'azienda sarà dato da:

$$\sum_{j=1...J} \sum_{t=1...T} [Cext_{jt} * Vext_{jt}]$$

Questi costo si aggiungono alla funzione obiettivo da minimizzare (e 3.19) rendendola pari a:

$$Min! \quad \Sigma_{j=1..J} \Sigma_{t=1..T} [h_j * I_{jt}] + \Sigma_{j=1..J} \Sigma_{t=1..T} [sc_j * (Y_{jt} - W_{jt})] + \\ + \Sigma_{m=1..M} \Sigma_{t=1..T} [oc_{mt} * O_{mt}] + \sum_{j=1..J} \sum_{t=1..T} [Cr_{jt} * Vrit_{jt}] + \\ \sum_{j=1..J} \sum_{t=1..T} [Cext_{jt} * Vext_{jt}]$$

l'equazione del vincolo di bilancio (e 3.20) si modifica come segue:

$$I_{jt-1} + X_{jt} + Vrit_{jt} + Vext_{jt} + U_{jt} = P_{jt} + \Sigma_{k \in Sj} (r^d_{jk} * X_{kt}) + I_{jt}$$
$$\forall j = 1,...,J; t = 1,...,T$$

inoltre si aggiunge un vincolo relativo alla quantità di prodotto che j che + possibile esternalizza:

$$Vext_{jt} <= c\ P_{jt}$$ dove $c$ è un valore tra $[0,1]$

Al fine di consentire una pianificazione che tenga conto del flusso di materiale proveniente dalla riverse logistic, è necessario integrare i due algoritmi. Note le domande di prodotto per i vari periodi ($P_{jt}$) si lancia il CLSPL per fare una prima pianificazione fissando gli $U_{jt}=0$. I valori $X_{jt}$ ottenuti vengono utilizzati come valori di domanda nell'algoritmo di disassemblaggio selettivo proposto da Gupta. A valle dell'elaborazione di tale modello con l'ulteriore vincolo che la funzione obiettivo non deve risultare negativa (in tal caso il componente proveniente da prodotti a fine vita costerebbe di più del medesimo componente nuovo) si determina il flusso che effettivamente può provenire da prodotti EOL per ogni periodo. Tale flusso viene inserito nel CLSPL assegnando i valori ottenuti alle variabili $U_{jt}$ e quindi viene lanciato per la seconda volta l'algoritmo CLSPL.
L'implementazione degli algoritmi è stata effettuta con il supporto del software LINGO.

## 3.8. Sviluppo di una piattaforma di simulazione

Nell'ambito dei sistemi integrati di simulazione è stata implementata una piattaforma di simulazione, sviluppata mediante il software ARENA, con un metamodello grazie al quale, mediante la parametrizzazione di una base di dati, si genera in maniera veloce modelli di processi di produzione di beni/servizi. Inoltre, le politiche gestionali di routing, scheduling, sequencing etc, vengono genericamente indicate mediante una funzione F(X) ove X rappresenta una matrice di variabili di stato del processo durante la simulazione. In tal senso si parla di piattaforma di simulazione, in quanto anziché stabilire a priori delle funzioni che svolgano i diversi compiti sopra citati, si desidera poter sfruttare tecnologie e modelli, eventualmente realizzati da altri, ed integrarle nella piattaforma di simulazione. Tecnicamente, tale integrazione può avvenire mediante l'utilizzo del protocollo SOAP. La piattaforma di simulazione implementata può essere schematizzata come in Figura 3.6. E' evidente che il solo sviluppo della piattaforma richiede uno sforzo significativo e lo stato dell'arte evidenzia un limitato numero di approcci di questo genere e pertanto già questo

dà una connotazione di innovatività; ma la vera innovatività sta nello sfruttare al meglio le potenzialità di uno strumento del genere. Una volta ottenuto il modello di simulazione è possibile effettuare una serie di esperimenti per cercare di trovare la configurazione migliore delle variabili controllabili del processo produttivo oppure valutare modifiche al piano di produzione che possano comportare anche eventuali backlog ovvero ordini inevasi entro la data promessa di consegna. A valle della simulazione è possibile effettuare anche una valutazione dei costi collegati alla configurazione scelta e quindi selezionare la configurazione migliore. In quest'ottica le tecniche di progettazione degli esperimenti e analisi della varianza rappresentano sia da un punto di vista teorico che da un punto di vista applicativo la parte principale dell'attività di ricerca.

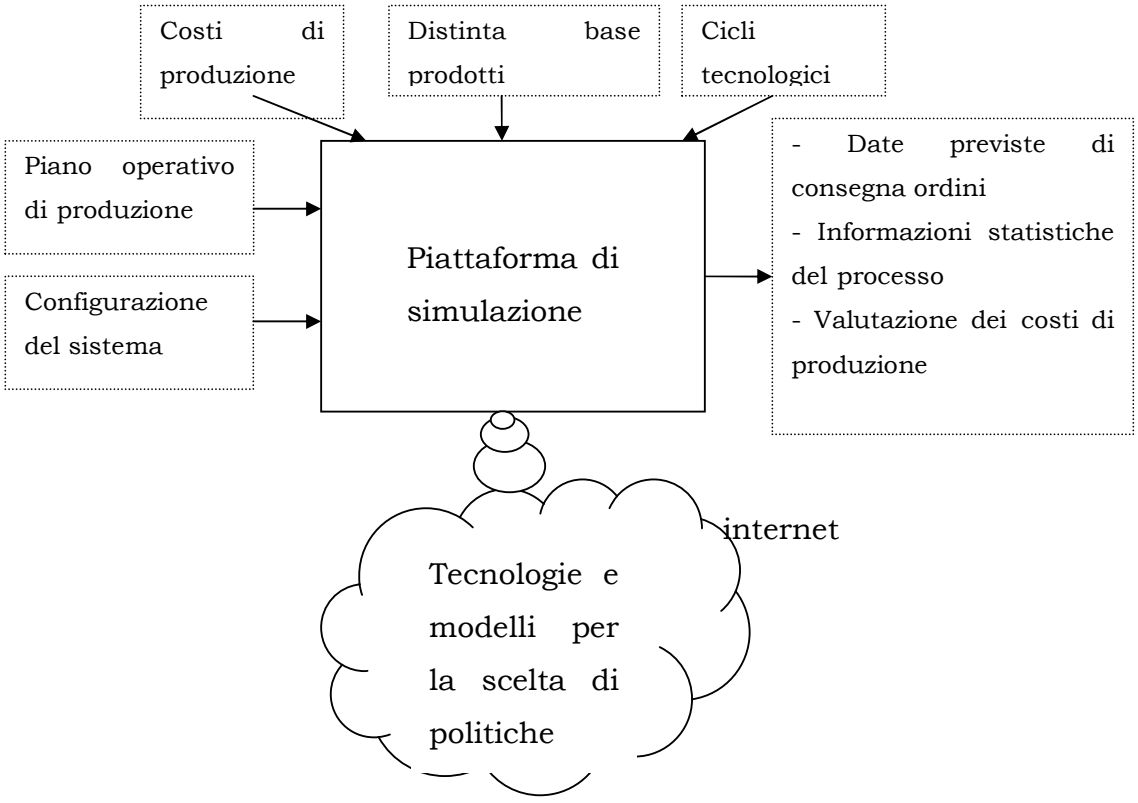


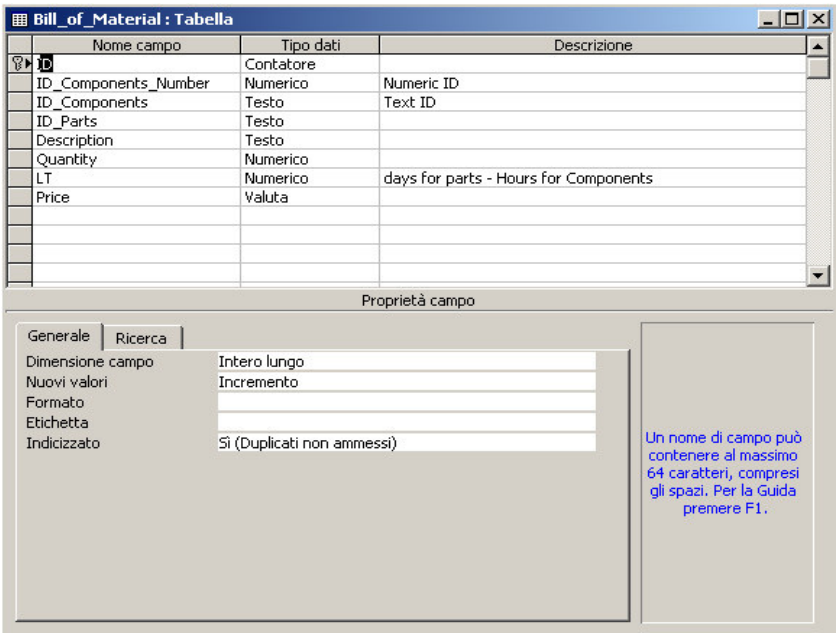

**Figura 3.6**

Per la schematizzazione del modello di processo produttivo è stata realizzata una base dati che dovrà contenere tutte le informazioni per la parametrizzazione e la registrazione dei risultati derivanti dalla piattaforma di simulazione. In particolare la base dati è composta dalle seguenti tabelle:

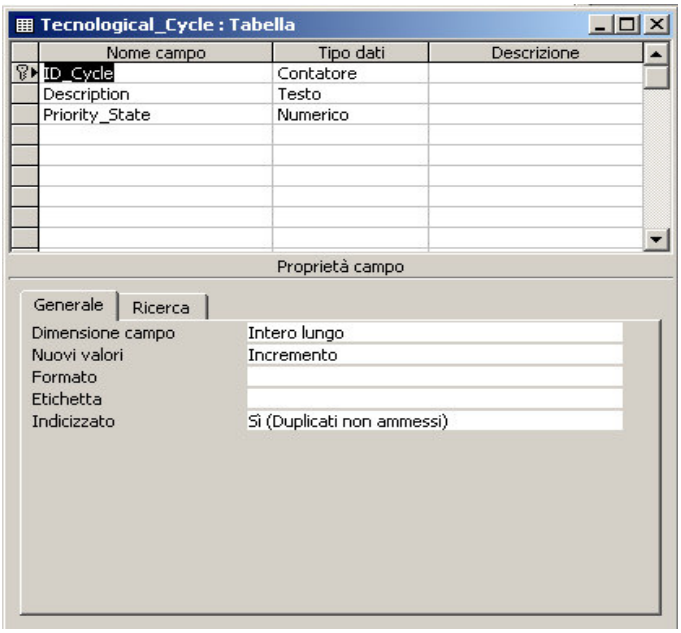
| Tipologia tabella | Nome Tabella | Descrizione |
|---|---|---|
| Tabelle di Produzione | Bill_of_Materials | Distinta base |
| | Tecnological_Cycle | Definisce i cicli tecnologici |
| | Tecnological_Cycle_Sequence | Definisce la sequenza delle lavorazioni da effettuare dato un certo ciclo tecnologico |
| | Work | Defisce le lavorazioni che possono essere associate ad un ciclo tecnologico |
| | Resource | Definisce l'insieme delle risorse (in termini strumentali: macchine, strumenti di misura etc) disponibili nell'impianto. |
| | Resource's_Works | Definisce le risorse necessarie per ciascuna lavorazione |
| | Other_Resource | Definisce altre risorse necessarie per il funzionamento di una risorsa strumentale (es. operatori, attrezzi etc) |
| | Other_Resource_for_Resource | Abbina le "Other Resource" alle risorse strumentali |
| | Required_Material_in Cycle_Phases | Definisce i materiali richiesti per la realizzazione di un determinato ciclo tecnologico (es. materiali di consumo). |
| | Relationship_Cycle_Component | Definisce quali cicli tecnologici alternativi possono essere utilizzati per realizzare un dato componente (composto da diverse parti). |
| | Setup_Time | Definisce il tempo di setup di una risorsa per passare da una lavorazione ad un'altra |
| | Logic_Stores | Magazzini logici assegnati a ciascun ordine di produzione (il magazzino 0 è il magazzino generale) |
| Tabelle Ordini | Production_Order | Ordini di produzione |
| | Purchase_Order | Ordini di approvvigionamento |
| Altre Tabelle | Calendar | Definisce una corrispondenza tra i giorni lavorativi e l'ora di simulazione. |
| | Weekly_Hours_Job | Definisce il numero di ore di lavoro in corrispondenza dei giorni della settimana. In base a tale tabella una procedura automatica genera la tabella Calendar. |
| | Events_Register | Il registro degli eventi consente di monitorare i batch d'interesse durante l'attraversamento dei diversi step della simulazione. |
| | Measure_Units | Definisce le diverse unità di misura utilizzate nei diversi registri/tabelle |
| | Simulation_Register | Il registro di simulazione memorizza alcuni dati delle simulazioni effettuate. |
| Registri di Magazzino e Contabilità | Master_Economic_Register | Il registro di contabilità ha la funzione di memorizzare tutti gli eventi che determinano una spesa/ricavo. |
| | Master_Store_Register | Il registro di magazzino ha la funzione di memorizzare tutti i movimenti di |

| | | |
|---|---|---|
| | | carico/scarico dai diversi magazzini logici. |
| | Type_of_Movement | Definisce le diverse tipologie di movimento possibili (es. Costo/Ricavo, Carico/Scarico, etc.) |

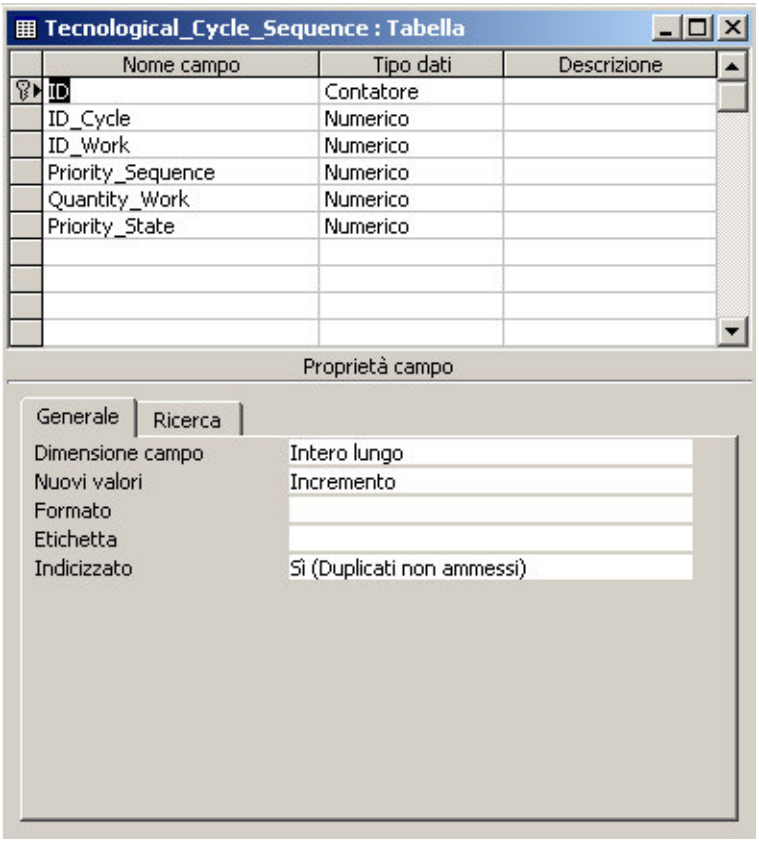
Nel dettaglio le tabelle sono composte dai campi in figura:




Oltre al campo identificativo del record (ID), in tabella sono presenti i campi ID_Components_Number ed ID_Components che rappresentano una codifica numerica ed alfanumerica delle Parti e dei Componenti (assiemi di parti). ID_Parts, invece, rappresenta il codice alfanumerico dell'assieme cui una componente appartiene. Description rappresenta una descrizione del componente/parte. Quantity rappresenta il coefficiente d'impiego del componente nella parte (es. 1 parte A è composta da 2 componenti B ed 1 componente C). LT rappresenta il lead time medio di approvvigionamento/produzione. Price è il prezzo di acquisto/costo di produzione.

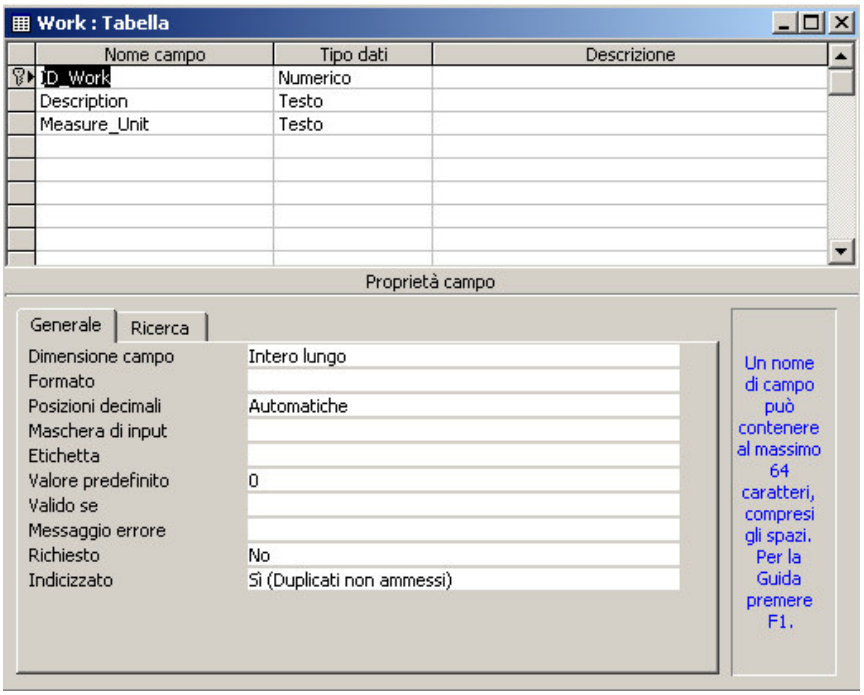



Il campo ID_Cycle identifica univocamente un ciclo tecnologico. Il campo description ne da una breve descrizione. Il campo priority state varia durante la simulazione a seconda delle condizioni che si verificano. Maggiore è il priority state più probabile risulterà la realizzazione di un certo prodotto attraverso tale ciclo tecnologico (se è sono presenti diversi cicli tecnologici alternativi per la realizzazione del prodotto).

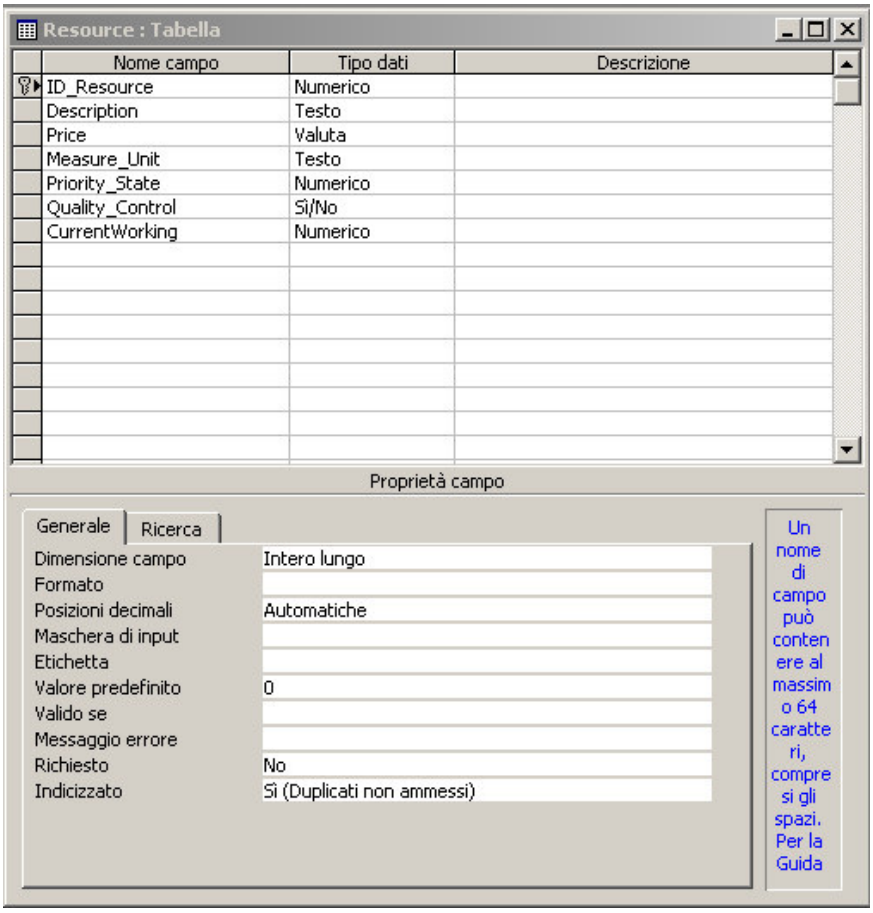


Oltre al campo ID che identifica univocamente il record, troviamo il campo ID_Work che rappresenta una delle lavorazioni di cui è composto il ciclo tecnologico ID_Cycle. La Priority_Sequence definisce la sequenza con cui devono essere eseguite le lavorazioni. Qualora due lavorazioni abbiano la medesima Priority_Sequence, significa che possono essere eseguite indifferentemente prima l'una e poi l'altra. In tal caso verrà scelta quella avente il Priority_State maggiore. Tale campo verrà determinato durante la simulazione in funzione dello stato del sistema. Quantity_Work rappresenta una "quantità di lavorazione" necessaria, tale quantità a seconda dell'unità di misura utilizzata verrà tramutata in un tempo di lavorazione.

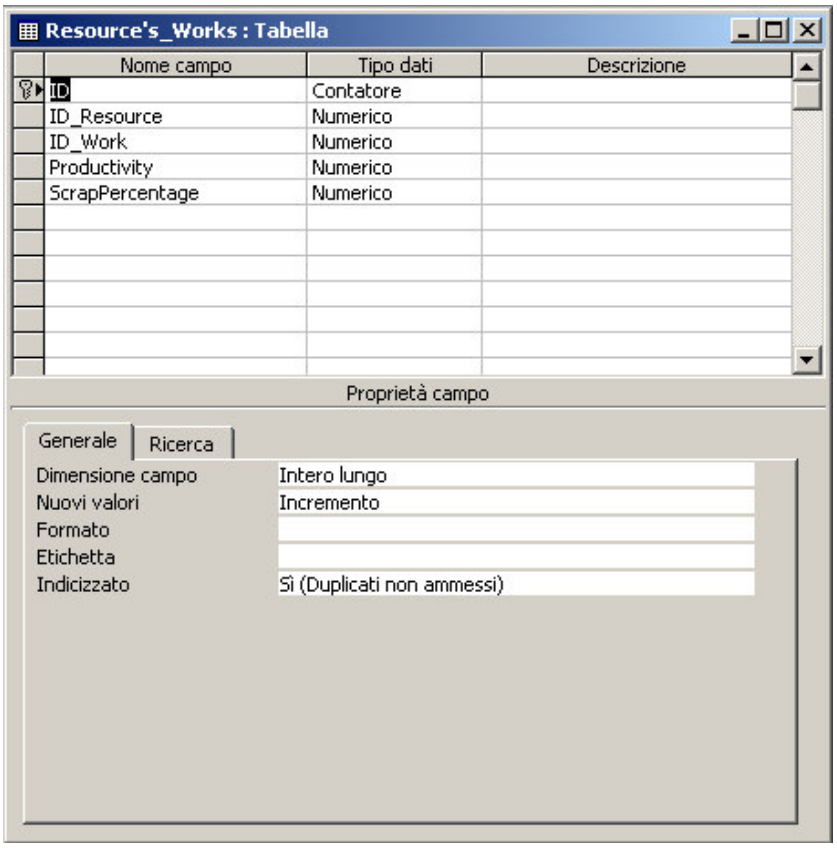
.



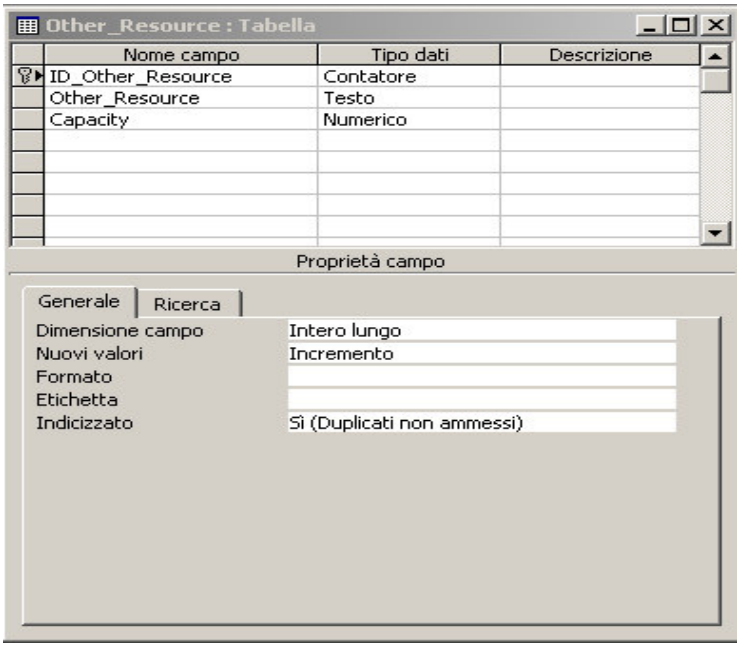

ID_Work è un codice identificativo della lavorazione. Description è una descrizione della lavorazione, Measure Unit è l'unità di misura utilizzata nella lavorazione.



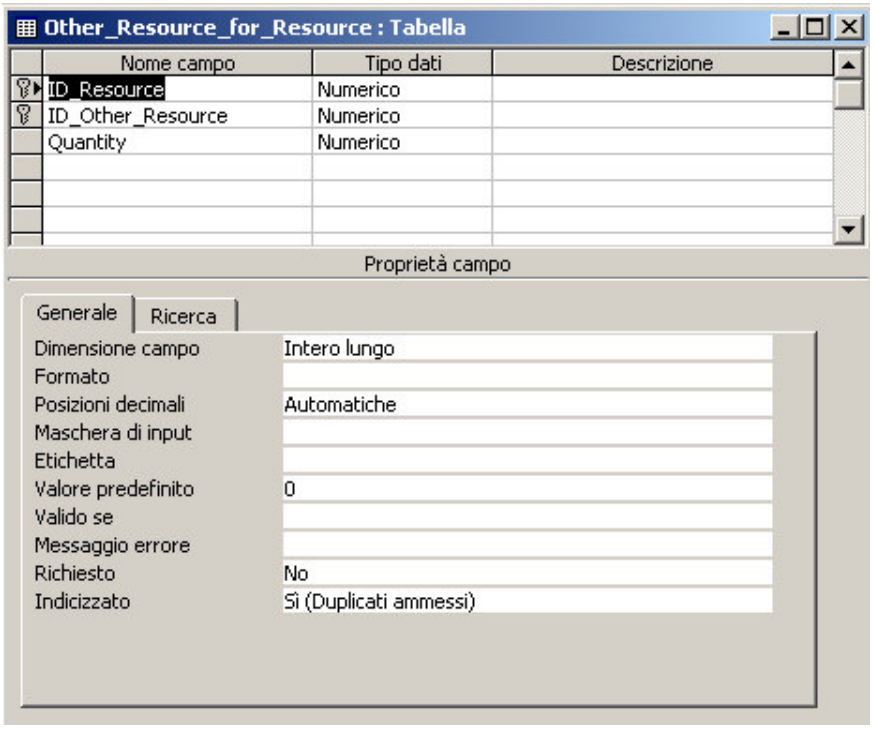

ID_Resource è un indice identificativo delle risorse disponibili. Description rappresenta una descrizione della risorsa. Price è il costo della risorsa nell'unità di misura espressa attraverso Measure_Unit. Priority_State rappresenta un indice che è funzionale allo stato del sistema. Qualora per la realizzazione di una lavorazione possano essere utilizzate diverse risorse, quella avente priority state maggiore sarà quella utilizzata.



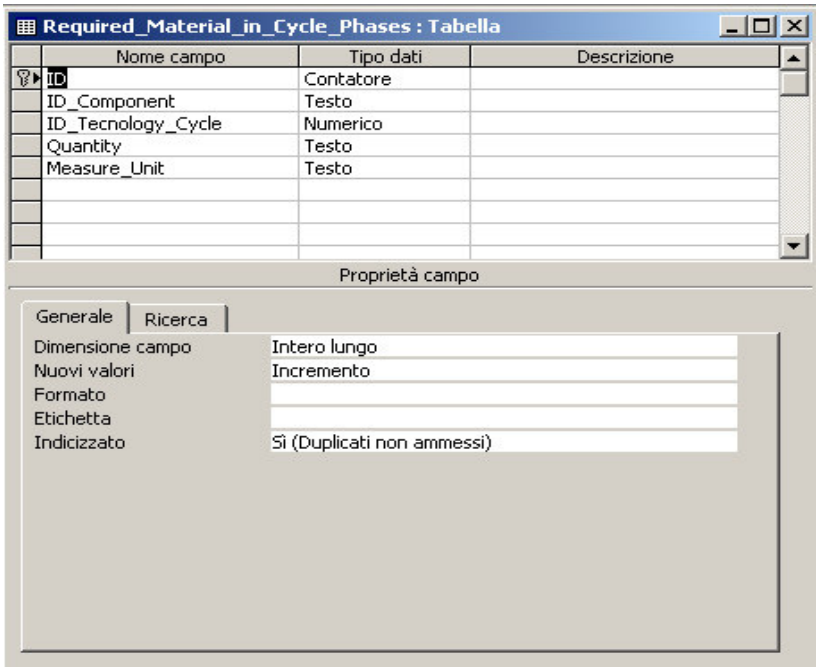

Il campo ID identifica univocamente i record della tabella. ID_Resource rappresenta la risorsa associata alla lavorazione ID_Work. Productivity è la produttività della risorsa (in termini di pezzi/min ad esempio). ScrapPercentage è la percentuale di scarto della lavorazione ID_Work effettuata con la risorsa ID_Resource.



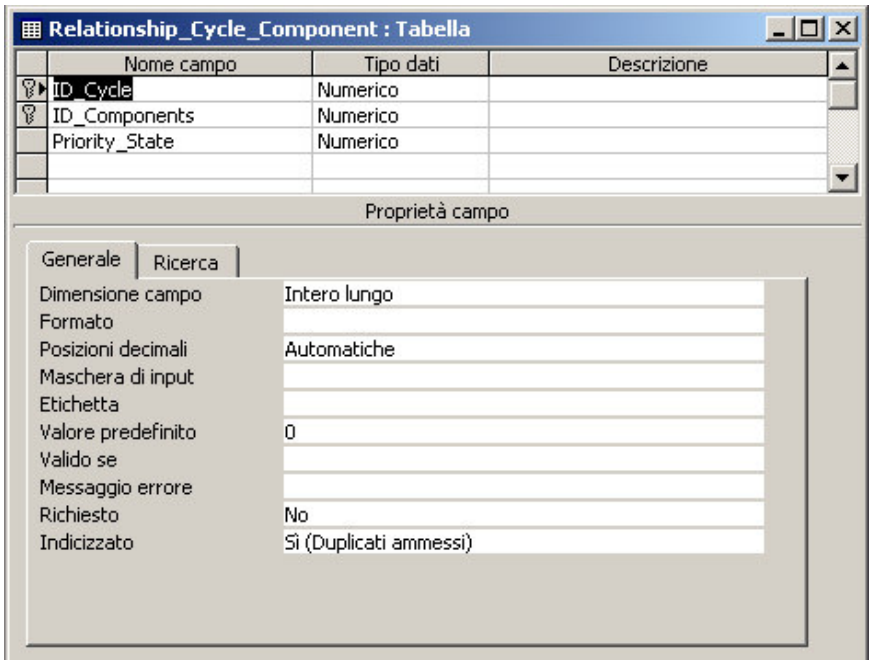
ID_Other_Resource è un identificatore di ciascuna "other resource" presente nell'impianto. Other_Resource è una descrizione della risorsa (es. Operatore di saldatura). Capacity è la quantità di risorsa di disponibile.



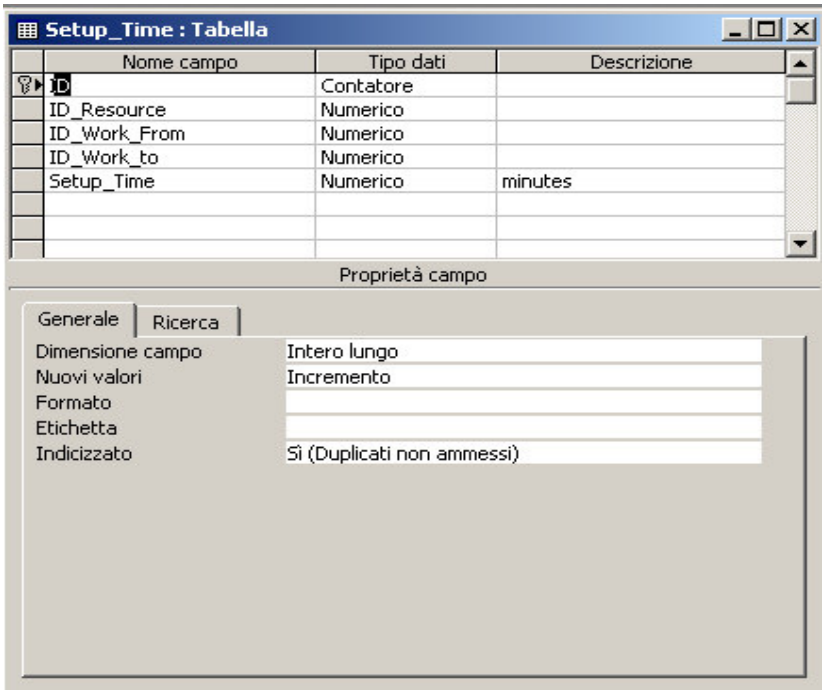

ID_Resource rappresenta la risorsa cui necessita ID_Other_Resource per poter funzionare. Quantity rappresenta la quantità di ID_Other_Resource necessaria a ID_Resource.



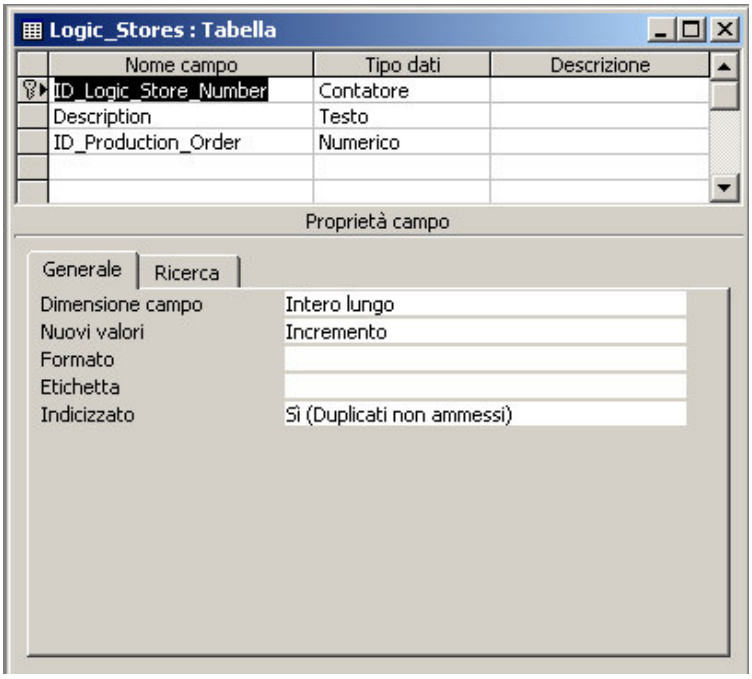
ID rappresenta un campo identificativo del record. ID_Component rappresenta il codice del materiale richiesto per il ciclo tecnologico ID_Tecnology_Cycle. Quantity rappresenta la quantità di ID_Component richiesta. Measure_Unit indica l'unità di misura di ID_Component.



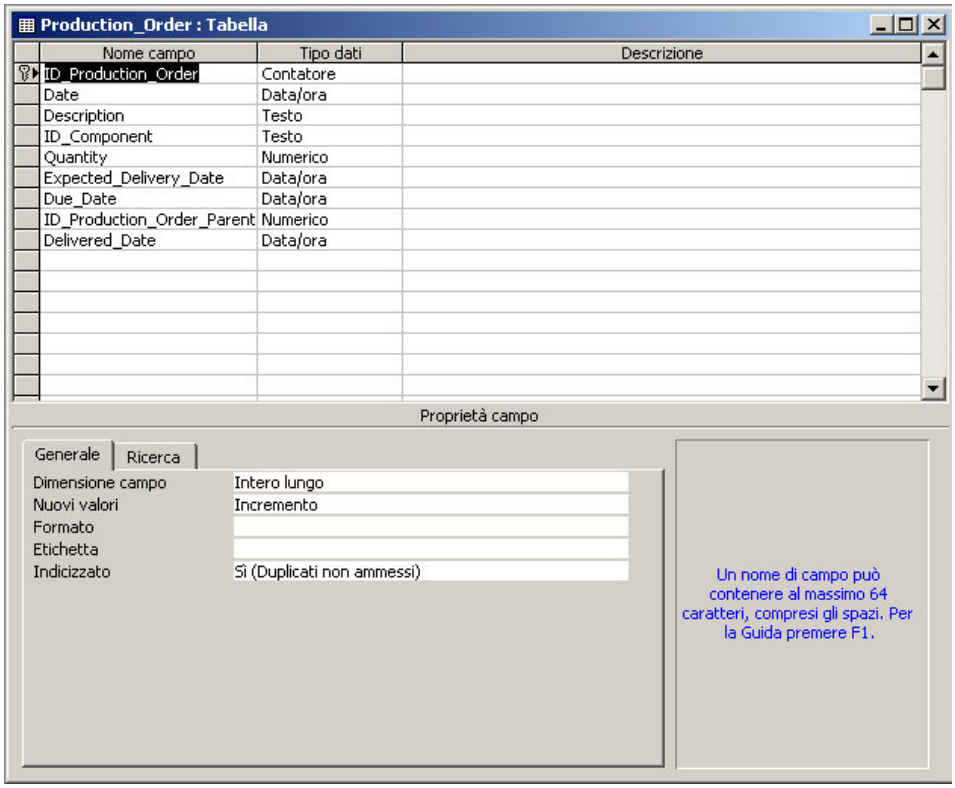

ID_Cycle indica il ciclo tecnologico che è possibile utilizzare per realizzare ID_Components. Priority_State è funzione dello stato del sistema durante la simulazione e serve a stabilire tra diversi cicli tecnologici alternativi quale scegliere.



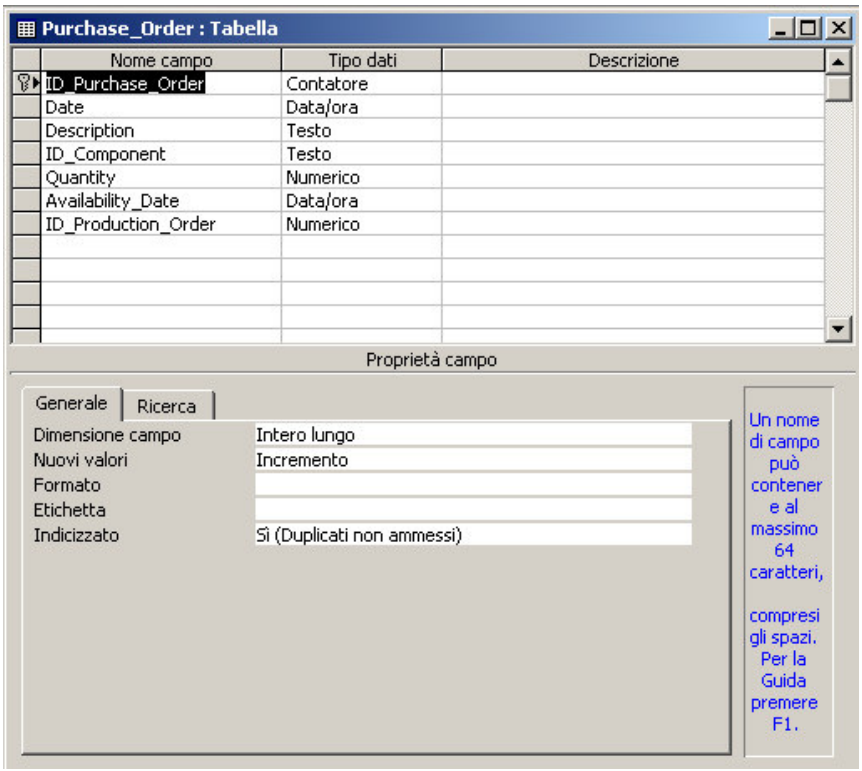
ID è un codice identificativo del record. Setup_Time definisce il tempo di setup per passare dalla lavorazione ID_Work_From alla lavorazione ID_Work_to sulla risorsa ID_Resource.



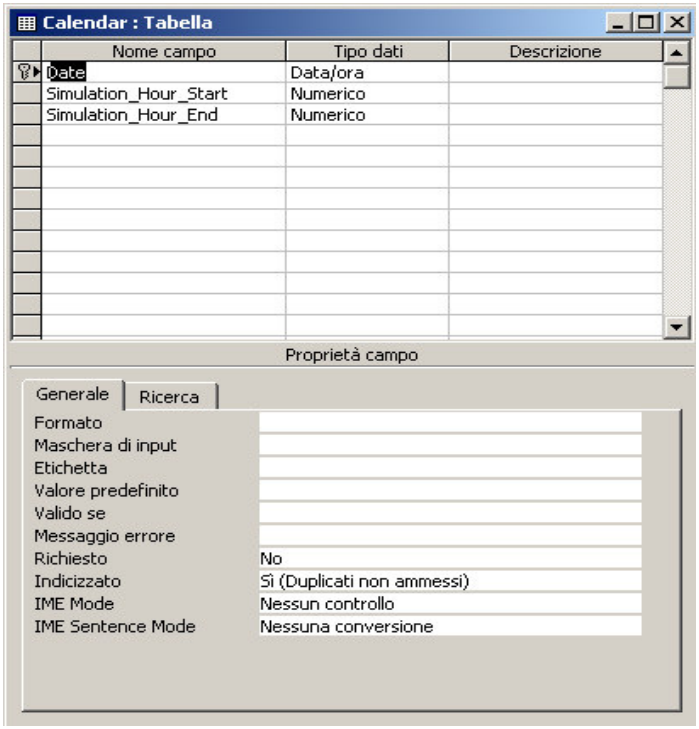

ID_Logic_Store_Number identifica univocamente il magazzino logico. Description fornisce una descrizione di tale magazzino. ID_Production_Order definisce a quale ordine di produzione/commessa è legato il magazzino logico.



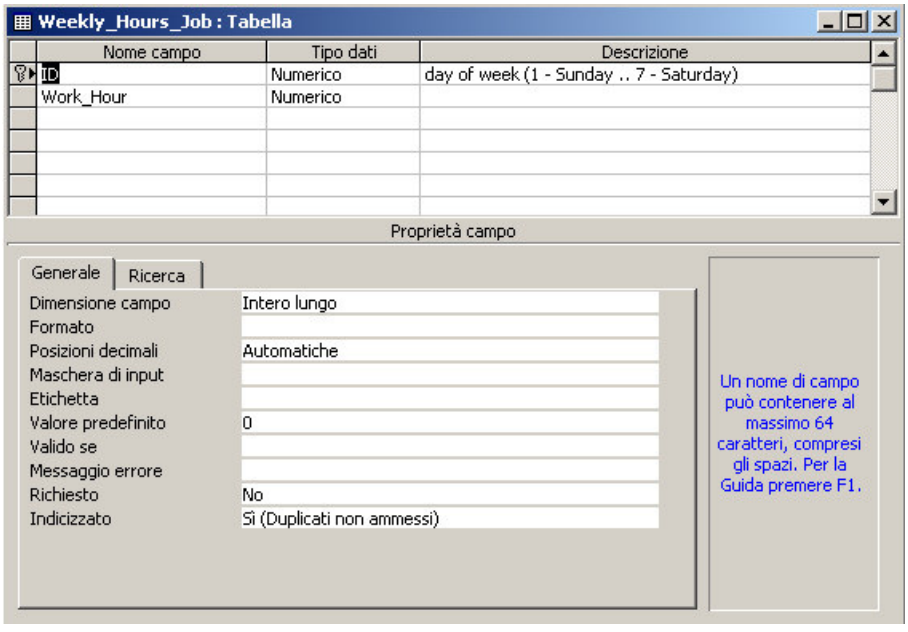
ID_Production_Order identifica univocamente l'ordine di produzione. Date rappresenta la data in cui l'ordine viene emesso. Description è una descrizione dell'ordine. ID_Component rappresenta l'assieme da consegnare. Quantity rappresenta la quantità richiesta. Expected_Delivery_Date è la data prevista di consegna. Due_Date è la data ultima entro cui consegnare. Delivered_Date è la data in cui l'ordine è stato consegnato. ID_Production_Order_Parent è l'eventuale ordine di produzione che ha determinano il lancio dell'ordine di produzione ID_Production_Order.



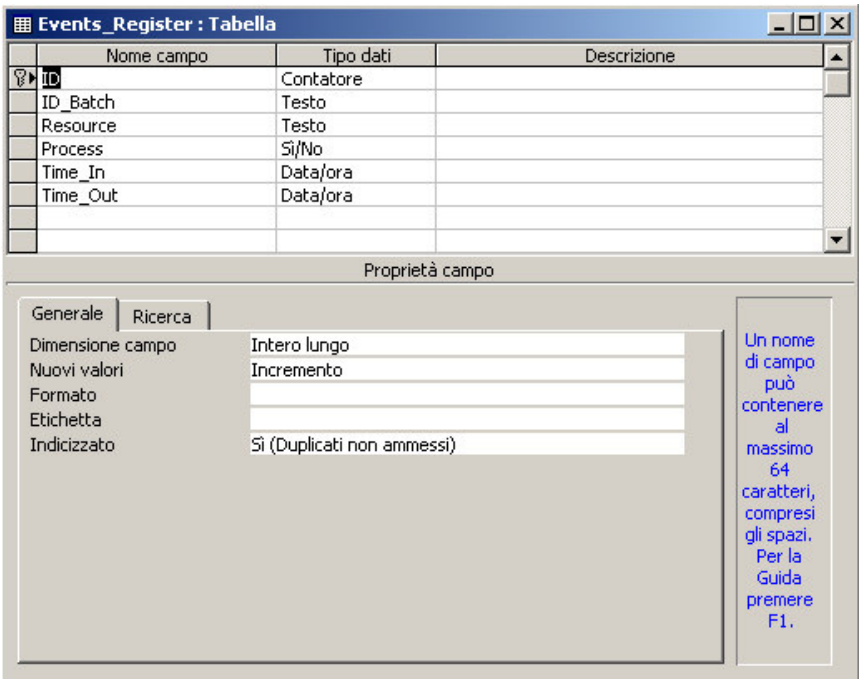

ID_Purchase_Order indentifica univocamente l'ordine di approvvigionamento. Date indica la data di lancio ordine. Description rappresenta una descrizione dell'ordine. ID_Component indica il componente richiesto. Quantity indica la quantità richiesta. Availability_Date è la data in cui di presume sia disponibile il componente. ID_Production_Order è l'ordine di produzione che ha determinato il lancio dell'ordine di acquisto.



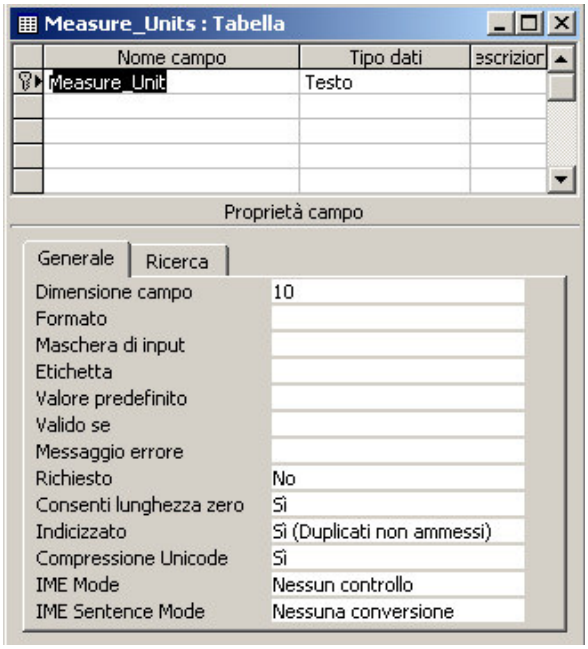

Date indica il giorno dell'anno (lavorativo). Simulation_Hour_Start e Simulation_Hour_End rappresentano rispettivamente l'ora simulata iniziale e l'ora simulata finale relativa al giorno Date.



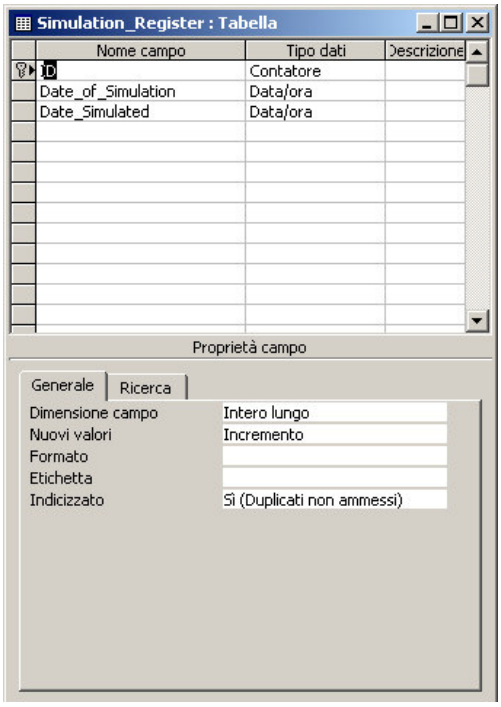

ID indica, mediante un numero da 1 a 7, il giorno della settimana. Work_Hour indica quante ore si lavora in ciascun giorno.



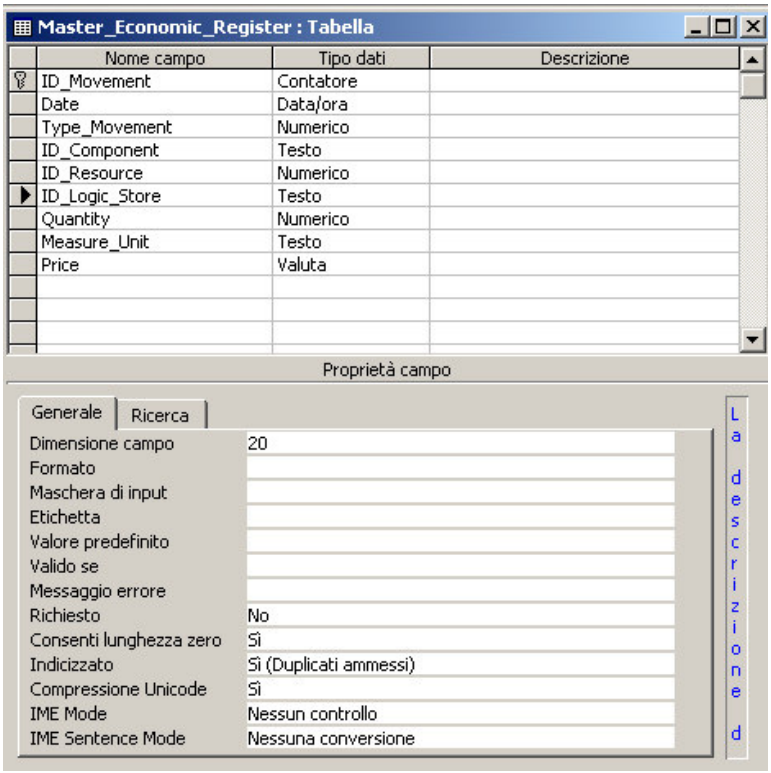

ID è un codice identificativo del record. ID_batch rappresenta il codice del lotto che si è monitorato. Resource è la risorsa che in tempo reale viene utilizzata dal batch. Process indica se la risorsa utilizzata è abbinata ad una lavorazione. Time_In e Time_Out rappresentano rispettivamente l'ora di ingresso e di uscita dalla risorsa/coda.




Measure_Unit indica le unità di misura utilizzate nel modello.

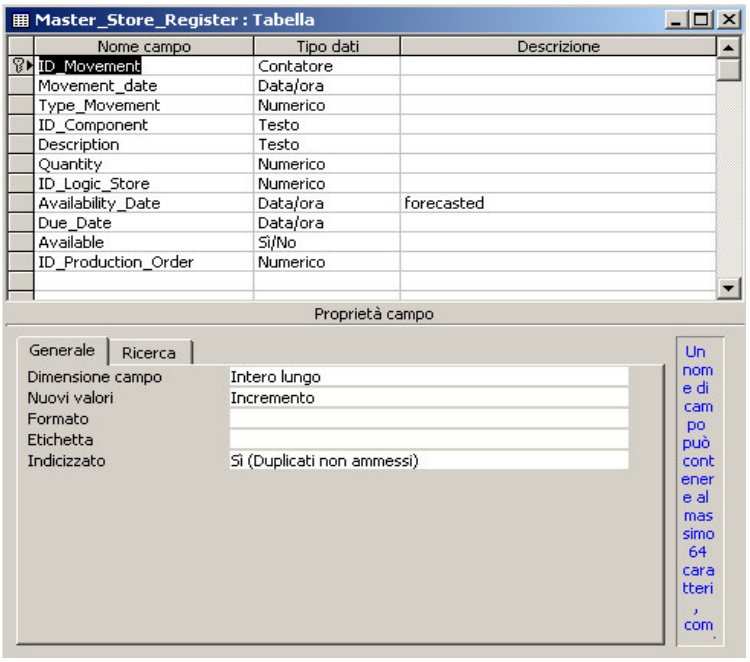



ID è un campo identificativo del record. Date_of_Simulation è la data di lancio della simulazione. Date_Simulated è la data simulata (data di partenza della simulazione).

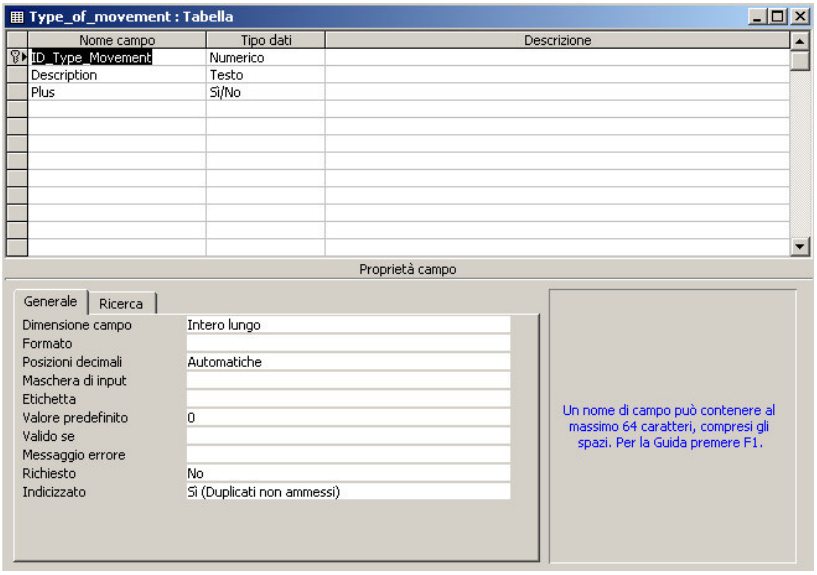



ID_Movement è un codice identificativo del movimento contabile. Date è la data della registrazione contabile. Type_Movement indica se il movimento è un costo o

un ricavo. ID_Component è il componente/assieme che ha, eventualmente, generato il movimento. ID_Resource è la risorsa che ha, eventualmente, generato il movimento. ID_Logic_Store è il magazzino logico che ha, eventualmente, generato il movimento. Quantity rappresenta la quantità in termini di unità di misura Measure_Unit che ha determinato il costo/ricavo Price.

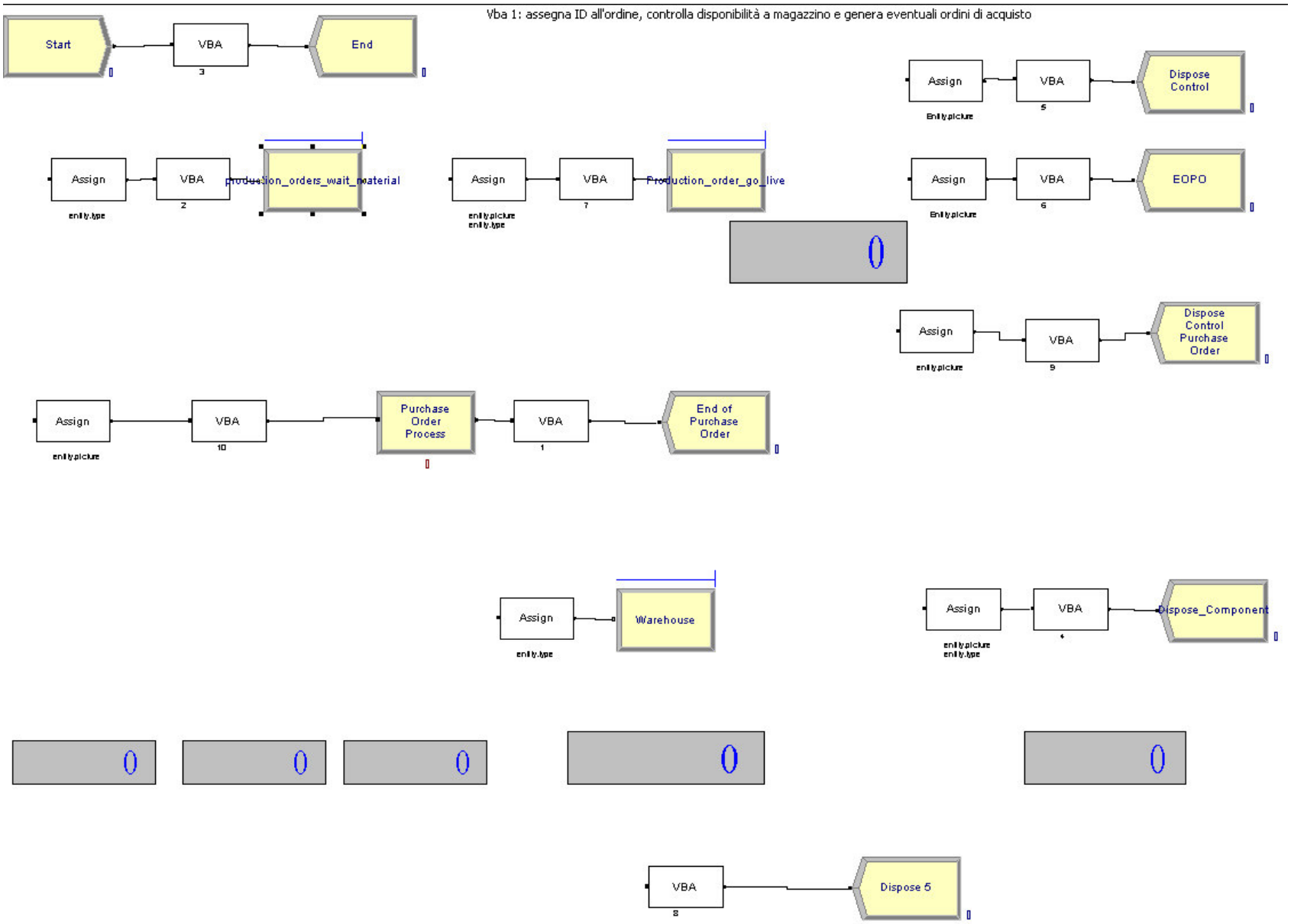



ID_Movement è il codice identificativo del movimento di magazzino. Movement_data indica la data del movimento. Type_Movement indica se il movimento è di carico/scarico. Description rappresenta una descrizione del movimento. Quantity rappresenta la quantità da movimentare in magazzino. ID_Logic_Store indica il magazzino logico interessato dalla movimentazione. Availability_date rappresenta la data in cui si prevede che risulterà disponibile il componente/assieme movimentato. Due_Date è la data entro cui dovrà essere disponibile il componente/assieme in magazzino. Available è una variabile booleana che indica se è disponibile fisicamente il componente/assieme. ID_Production_Order indica quale ordine di produzione ha generato il movimento.




ID_Type_Movement è un codice identificativo del tipo di movimento. Description rappresenta una descrizione del movimento. Plus indica se un movimento è positivo (es. Carico/Ricavo) o negativo (es. Scarico/Costo).

Oltre alla base dati è stato realizzato il metamodello di simulazione (mostrato in figura). La simulazione comincia con il blocco Start. Tale blocco determina l'attivazione di una procedura che genera il modello del processo di produzione leggendo dal database tutte le risorse disponibili, i cicli di lavoro, le risorse necessarie per le lavorazioni ed altro. Inoltre, mediante tale procedura vengono letti gli ordini di produzione dal database ed inviati alla coda Production_orders_wait_material. In questo passaggio l'ordine attraversa una blocco VBA (2) che effettua una serie di verifiche applicando fasi principali dell'MRP (Material Requirement Planning).


Vba 1: assegna ID all'ordine, controlla disponibilità a magazzino e genera eventuali ordini di acquisto
Start
VBA
End
Assign
VBA
Dispose Control
Assign
VBA
production_orders_wait_material
Assign
VBA
Production_order_go_live
Assign
VBA
EOPO
0
Assign
VBA
Dispose Control Purchase Order
Assign
VBA
Purchase Order Process
VBA
End of Purchase Order
Assign
VBA
Warehouse
Assign
VBA
Dispose_Component
0
0
0
0
0
VBA
Dispose 5

Qualora in magazzino (blocco Warehouse) fossero disponibili i prodotti richiesti nell'ordine, quest'ultimo verrebbe immediatamente evaso e concluderebbe il suo ciclo di vita. Se invece fosse necessario lanciare la produzione del prodotto ordinato bisognerà verificare la disponibilità dei componenti (effettuando una esplosione della distinta base).
Se i componenti sono disponibili l'ordine passa in Production_Order_go_live e verranno inviati i materiali al modello del processo di produzione generato in precedenza. Quando le parti prodotte concludono il ciclo di produzione, i pezzi vengono caricati in magazzino ed l'ordine passa nello stato End_of_Production_Order. Nel database tale ordine risulterà delivered.
Se non sono disponibili tutti i componenti, verranno effettuati eventuali ordini di acquisto ed ordini di produzione di subassiemi. Pertanto l'ordine che li ha lanciati rimarrà nello stato production_order_wait_material fino a quando tutti i materiali risulteranno disponibili a magazzino. Periodicamente, quindi, viene effettuato una verifica grazie ad un'entità di controllo che attraverso il VBA (5) effettua un monitoraggio dei materiali presenti in magazzino e degli ordini in attesa di materiali. Quando tutti i materiali sono disponibili l'ordine passa in go_live e seguirà lo stesso percorso citato in precendenza.
Il modello di processo produttivo generato dal metamodello di simulazione prevede sia stazioni di lavorazione che stazioni controllo qualità. Inoltre sono state implementate delle regole di routing dei pezzi tra le macchine grazie ai Priority_State presenti nel database. In particolare tali regole potrebbero essere implementate mediante altri sistemi (reti neurali, altre simulazioni etc.) in maniera completamente trasparente al metamodello che fornisce le variabili di stato della simulazione e riceve un valore del Priority_State.

## 3.9. Le possibili evoluzioni dell'approccio proposto

L'impostazione comune che si da alla pianificazione della produzione è mirata, per lo più, ad una riduzione dei costi e dei tempi. Un criterio diverso potrebbe essere quello di cercare, una soluzione che non necessariamente comporti costi inferiori ma risulti, altresì, più robusta.

In questo senso potrebbe essere di particolare interesse uno studio delle variabilità connesse ad un processo di produzione. Tali variabilità non sono, spesso, schematizzabili come semplici variabili aleatorie. Queste ultime sono normalmente gestite dai software di simulazione senza difficoltà. Il problema è che nel momento in cui si desidera gestire variabili aleatorie che non risultino stocasticamente indipendenti dagli eventi accaduti in altri periodi, oppure variabili aleatorie i cui parametri cambiano nel tempo, i sistemi di simulazione vanno opportunamente personalizzati. In tal senso, per l'impostazione data alla piattaforma sviluppata, è possibile realizzare simulazioni di processi in cui le variabili aleatorie abbiano un comportamento come quello descritto.

In generale si parla di funzioni aleatorie di una variabile deterministica *t* indicando una funzione i cui punti sono realizzazioni di altrettante variabili aleatorie. In particolare si definisce processo stocastico una funzione aleatoria in cui la variabile t è continua [62].

Nelle applicazioni industriali questo concetto è di estremo interesse, in quanto ci sono dei fenomeni che, pur avendo una bassa probabilità assoluta di accadimento, potrebbero presentare, nel tempo, una probabilità condizionata significativamente diversa. In tal modo, infatti, è possibile spiegare perché, talvolta, processi produttivi, inizialmente in controllo statistico, senza l'intervento di cause speciali, manifestino fenomeni di fuori controllo. Tali fenomeni sono dovuti a *segnali deboli* di cui non si tiene conto in fase di pianificazione della produzione essendo la loro rilevazione difficile e costosa.

Un indicatore molto utilizzato nella pratica operativa, per cercare di percepire segnali deboli, e non solo, che possono portare il sistema alla deriva, è il FOR (Fall Off Rate) definito come:

$$F.O.R.\% = \frac{D*100}{N}$$

dove D è il numero dei difetti riscontrato nel corso del processo produttivo e N è il numero di prodotti realizzati.

Da interviste effettuate a responsabili di stabilimento e responsabili di produzione è risultato che solitamente le cause che determinano tali difettosità possono classificarsi in tre gruppi:

- Manodopera (50%);
- Processo produttivo (20%);

- Qualità dei materiali (30%).

Giova evidenziare che le aziende intervistate hanno linee di produzione flowshop, con processi produttivi molto automatizzati e quindi la difettosità legata al processo produttivo dipende, principalmente, dalla qualità dei materiali e dalla politiche manutentive. Qualora dovessimo considerare un processo produttivo di tipo jobshop è naturale che le percentuali indicate subirebbero significativi mutamenti.

Dalle informazioni raccolte risulta piuttosto chiaro come un cospicuo numero di cause di deriva del sistema siano legate alla manodopera. Alcune cause frequenti sono relative a:

- negligenze di vario genere
- stanchezza dell'operatore specie verso la fine del turno di lavoro
- errori nel trasferimento di informazioni

Per quanto concerne la prima causa indicata, essa risulta completamente casuale e non si riesce, nella maggior parte dei casi, a comprendere le leggi che ne governano l'evoluzione.

La seconda causa indicata potrebbe essere assimilata ad un normale processo di usura e, quindi, schematizzabile con i modelli tradizionalmente utilizzati per i componenti meccanici.

In merito agli errori nel trasferimento di informazioni, tale causa di difettosità/deriva del sistema potrebbe essere schematizzabile con una funzione aleatoria discreta avente sezione di tipo Bernoulliano [62]. Infatti detta $X_i$ una variabile di tipo binario in cui il valore 1 indica l'evento *informazione trasferita correttamente* ed il valore 0 indica l'evento *informazione trasferita non correttamente*, se

$$\Pr\{X_i = 1\} = 0.99$$
$$\Pr\{X_i = 0\} = 0.01$$

via via che viene trasferita un'informazione nelle varie fasi del processo produttivo tra i diversi operatori, la probabilità che all'i-esimo trasferimento l'informazione trasmessa sia corretta è:

$$\alpha(i) = \alpha(i-1) \cdot 0.99 + [1 - \alpha(i-1)] \cdot 0.01$$

Si può dimostrare che dopo appena 50 trasferimenti tale probabilità si riduce a circa 0,7.

Per quanto concerne le difettosità indotte dalla qualità dei materiali, si potrebbe pensare ad un modello in cui il materiale in ingresso possiede un certo grado di non-qualità. Tale livello non può mai diminuire in seguito alle lavorazioni, può solo aumentare o, al più, rimanere invariato. Se il grado di non-qualità aumenta oltre una certa soglia, il pezzo viene scartato. Quando il pezzo subisce una certa lavorazione, in funzione del grado di non-qualità posseduto dal pezzo e determinato dalle diverse lavorazioni precedenti, c'è una certa probabilità (condizionata) che la lavorazione possa determinare un aumento della non-qualità. Tale incremento risulta funzione del grado di non-qualità delle lavorazioni precedenti che sono stocasticamente correlate alla lavorazione corrente.
Se pensiamo ad una linea di produzione composta da una serie di m macchine che realizzano delle operazioni sul prodotto, risulta che:
detto $E_i$ l'evento "aumento del grado di non-qualità alla lavorazione i", $X_i$ il grado di non-qualità determinato dalla lavorazione i-esima (indipendentemente dalle non-qualità dovute ad altre lavorazioni) e $a_i$ un coefficiente di amplificazione che tiene conto delle non-qualità dovute ad altre lavorazioni,

$$\Pr\{E_1 \mid X_0\} = \alpha_1,\ \Pr\{E_2 \mid X_0, X_1\} = \alpha_2\ \ldots\ldots,\ \Pr\{E_m \mid X_0, X_1, .., X_m\} = \alpha_m$$

e

$$X_1 = a_0(X_0)\cdot X_0,\ X_2 = a_0(X_0)\cdot X_0 + a_1(X_0, X_1)\cdot X_1,\ \ldots,\ X_m = a_0(X_0)\cdot X_0 + \ldots + a_m(X_0, X_1, \ldots, X_m)\cdot X_m$$

E' evidente che la valutazione sul campo di tali correlazioni risulta piuttosto complessa, però i vantaggi che se ne potrebbero trarre sono certamente non trascurabili. Sia nei sistemi flow-shop che in quelli job-shop esistono sempre dei gradi di libertà nella sequenza tecnologica delle operazioni da effettuare per produrre un certo bene. È evidente che, note le correlazioni sopra citate, sarebbe possibile pensare ad una schedulazione che, modificando la sequenza tecnologica delle lavorazioni, renda qualitativamente migliore il prodotto senza modificare le macchine presenti sulla linea produttiva. In tal senso si può affermare che la schedulazione realizzata in quest'ottica tende a migliorare la qualità del prodotto organizzando diversamente le risorse, senza la necessità di ingenti investimenti. I maggiori vantaggi di tale approccio si manifestano, comunque, in sistemi produttivi di tipo job shop, in quanto i gradi di libertà del sistema sono significativamente maggiori.

# 4. Analisi dell'output

## 4.1. Introduzione

A causa dell'aleatorietà intrinseca dei componenti che costituiscono un modello di simulazione, gli output dello stesso sono anch'essi aleatori.
Una replica, o run, non è altro che una prova eseguita sul modello, della durata dipendente dalle caratteristiche del sistema reale allo studio, e basata su un set di input cui sono stati assegnati opportuni valori. Effettuando un numero di repliche $n$ maggiore di uno, si eseguono più prove, $n$ per la precisione, tutte della medesima durata, sul medesimo modello e fondate sugli stessi valori degli input.
Quando il modello si basa su input deterministici, non modificando alcuna informazione tra una replica e l'altra, gli output sono gli stessi, per cui, in tale circostanza, è inutile eseguire più repliche. Quando invece gli input, non necessariamente tutti, seguono delle distribuzioni statistiche, in generale, gli output sono essi stessi variabili aleatorie. L'errore tipico che si commette nell'effettuare una simulazione è quello di eseguire un singolo run e considerare i risultati che si osservano come stime del comportamento del sistema reale. Tali stime, in realtà, sono solo valori particolari di variabili aleatorie che possono anche avere varianza molto elevata. Affinché i risultati di uno studio effettuato attraverso simulazione abbiano senso è necessario l'uso di tecniche statistiche per progettare ed analizzare gli esperimenti di una simulazione. Tipicamente queste tecniche si basano sull'ipotesi che i dati da analizzare siano indipendenti e identicamente distribuiti, cosa che non si verifica per gli output di una simulazione.

Siano $Y_1, Y_2, \cdots$ i dati di output di un singolo run di una simulazione; ciascuna $Y_i$ può essere vista come una variabile aleatoria e quindi la collezione di variabili aleatorie $\{Y_i, i = 1,2,...\}$ è un processo stocastico. In generale, le variabili aleatorie $Y_i$ non sono né indipendenti né identicamente distribuite e, quindi, per l'analisi di

questi dati non possono essere applicati direttamente i metodi di analisi statistica. Per ovviare a questo inconveniente, si effettuano più repliche della simulazione, ciascuna di lunghezza m. Siano $y_{11}, y_{12},...y_{1m}$ la realizzazione delle variabili aleatorie $Y_1, Y_2,...Y_m$ ottenute con la prima replica. Nella seconda replica si avranno differenti realizzazioni delle variabili aleatorie $Y_1, Y_2,...Y_m$; siano esse $y_{21}, y_{22},...y_{2m}$. Effettuando n repliche indipendenti di lunghezza m, si ottiene:

$$\begin{matrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nm} \end{matrix} \qquad \text{(e 4.1)}$$

Le realizzazioni di una stessa replica non sono indipendenti ed identicamente distribuite, ma se per ogni $i = 1,2,...,m$, consideriamo le osservazioni $y_{1i}, y_{2i},...y_{ni}$, ovvero l'*i*-esima colonna della (e 4.1), allora esse costituiscono osservazioni indipendenti ed identicamente distribuite della variabile aleatoria $Y_i$. Quindi l'analisi statistica è applicabile alle osservazioni $y_{1i}, y_{2i},...y_{ni}$, per ogni fissato $i = 1,2,...,m$.

L'applicazione delle tecniche statistiche per l'analisi dell'output si differenzia in base al tipo di simulazione. Si possono distinguere:

- *simulazione con terminazione* (terminating simulation): è quella per cui esiste un evento naturale che specifica la durata di ciascuna replica, dettato dalle caratteristiche di interesse del sistema reale. In questo caso si è interessati al comportamento del sistema su un orizzonte temporale finito e alle condizioni iniziali, cioè le condizioni sotto le quali il sistema parte, hanno un ampio impatto sulle misure di prestazione del sistema.
- *la simulazione senza terminazione* (nonterminating simulation): è quella per cui non esiste un evento naturale che specifica la durata di ciascuna replica. Molti sistemi simulati in questo modo mostrano un comportamento stazionario, ovvero, in un run molto lungo, la distribuzione degli output è indipendente dal tempo nonché dalle condizioni iniziali [127].

Per capire meglio la differenza tra i due tipi di simulazione si analizzino lo stato stazionario e il transitorio.

Sia $\{Y_i, i = 1,2,...\}$ un processo stocastico che rappresenta i risultati di una simulazione e, per ogni $i = 1,2,...$, sia

$$F_i(y \mid I) = P(Y_i \le y \mid I) \qquad y \in \Re$$

la probabilità condizionata che l'evento $\{Y_i \le y\}$ accada, date le condizioni iniziali *I,* ovvero $F_i(\cdot \mid I)$ è la funzione di distribuzione di $Y_i$ date le condizioni iniziali *I.*

La $F_i(y \mid I)$ è detta anche *distribuzione transitoria* del processo di output al tempo (discreto) $i$ con condizioni iniziali *I.* In generale, la $F_i(y \mid I)$ è diversa per differenti valori di $i$ e per ogni insieme di condizioni iniziali *I.*

Se, per ogni $y$ e *I,*

$$\Pr\{Y_i \le y\} = F_i(y \mid I) \to F(y) = \Pr\{Y \le y\} \quad \text{per} \quad i \to \infty$$

allora $F(y)$ è la *distribuzione stazionaria* del processo di output $\{Y_i, i = 1,2,...\}$ e Y è la variabile aleatoria stazionaria di interesse con funzione di distribuzione $F(y)$.

Lo stato stazionario si raggiunge in teoria per $i \to \infty$. Nella pratica, tuttavia, esiste, molto spesso, un indice temporale finito *k* in corrispondenza del quale le distribuzioni rimarranno approssimativamente coincidenti. Quando questo accade, si assume che il sistema sia nello *stato stazionario* a partire dal tempo *k.* Inoltre se Y è una variabile aleatoria con distribuzione stazionaria *F,* allora $E(Y)$ è una misura di prestazione stazionaria e $E(Y_i \mid I) \to E(Y)$ per $i \to \infty$, per ogni insieme di condizioni iniziali $I$ [10].

## 4.2. Simulazione con terminazione: analisi del transitorio

Si supponga di effettuare n repliche indipendenti di una simulazione con terminazione, ciascuna delle quali termina all'occorrenza del medesimo evento "*E*" ed inizia con le stesse condizioni iniziali "*I*", o meglio caratterizzate dalla medesima distribuzione. Si assuma, per semplicità, che vi sia una singola misura di prestazione di interesse da voler stimare, rappresentata dalla variabile aleatoria $X$, e sia $X_j$ la realizzazione di questa variabile aleatoria ottenuta nella j-esima

replica, per $j = 1,2...n$. Per quanto detto in precedenza le $X_j$ sono v.a. indipendenti ed identicamente distribuite.

### *4.2.1. Stima della media e calcolo dell'intervallo di confidenza*

Si supponga di voler determinare la stima e l'intervallo di confidenza della media $\mu = E(X)$.

Uno stimatore corretto della media di una popolazione è dato da

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \text{(e 4.2)}$$

Esso è, inoltre, uno stimatore consistente, ovvero converge stocasticamente al parametro $\mu$ quando la dimensione $n$ del campione aumenta infinatemente: se, fissati un $\varepsilon$ e un $\delta$, positivi e comunque piccoli, per $n \to \infty$ vale la relazione

$$\Pr\left\{ \overline{X} - \mu \mid < \delta \right\} > 1 - \varepsilon \qquad \text{(e 4.3)}$$

Inoltre $\overline{X}$ è il migliore tra i tutti gli stimatori della media di una v.a., avendo la minima varianza [62].

Il teorema del limite centrale afferma che la somma di n v.a. s-indipendenti ed equidistribuite con media $\mu$ e varianza $\sigma^2$ tende, per $n \to \infty$, ad essere distribuita come una v.a. Normale con media $n\mu$ e varianza $n\sigma^2$. Poiché il numeratore della media campionaria $\overline{X}$ è la somma di n v.a. indipendenti ed identicamente distribuite con media $\mu$ e varianza $\sigma^2$, risulta che in virtù del teorema del limite centrale la Cdf della $\overline{X}(n)$ tende, al crescere di n, a quella Gaussiana di media $\mu$ e varianza $\sigma^2/n$, mentre la Cdf della corrispondente v.a. normalizzata

$$\frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \qquad \text{(e 4.4)}$$

tende a quella Gaussiana standard

$$\frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \overset{D}{\approx} N(0,1) \qquad \text{(e 4.5)}$$

dove il simbolo $\overset{D}{\approx}$ sta ad indicare che i due termini hanno approssimativamente la stessa distribuzione.

Fissato dunque un livello di confidenza $1-\alpha$, con $\alpha \in [0,1]$, l'intervallo di confidenza approssimato al $100(1-\alpha)\%$ è dato da

$$\overline{X}(n) \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}} \qquad \text{(e 4.6)}$$

Generalmente, nella maggior parte delle applicazioni, la varianza $\sigma^2$ non è nota ma è possibile sostituire nella (e 4.6) espressione il suo stimatore

$$S^2(n) = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X}(n))^2 \qquad \text{(e 4.7)}$$

Se le $X_i$ sono v.a. normali, la nuova v.a. $\dfrac{\overline{X} - \mu}{S/\sqrt{n}}$ è, per definizione, una v.a. t di Student con n-1 gradi di libertà, essendo il rapporto di un Gaussiana standard e la radice quadrata di una Chi-quadro $\chi^2$, divisa per i suoi gradi di libertà, s-indipendenti tra loro:

$$\frac{\overline{X} - \mu}{\frac{\sigma}{\sqrt{n}} \sqrt{(n-1)\frac{S^2}{\sigma^2} \Big/ (n-1)}} \qquad \text{(e 4.8)}$$

In tal caso, l'intervallo di confidenza approssimato al $100(1-\alpha)\%$ è dato da

$$\overline{X}(n) \pm t_{n-1,1-\frac{\alpha}{2}} \frac{S(n)}{\sqrt{n}} \qquad \text{(e 4.9)}$$

ovvero

$$\Pr\left\{ \overline{X}(n) - t_{n-1,1-\frac{\alpha}{2}} \sqrt{\frac{S^2(n)}{n}} \le \mu \le \overline{X}(n) + t_{n-1,1-\frac{\alpha}{2}} \sqrt{\frac{S^2(n)}{n}} \right\} \cong 1-\alpha .$$

Bisogna notare (Figura 4.1) che, per ogni n finito, la distribuzione t di Student è meno alta e più larga di quella normale, così che $t_{n-1,1-\frac{\alpha}{2}} > z_{1-\frac{\alpha}{2}}$, quindi l'intervallo di confidenza dato dalla (e 4.9) risulta più ampio di quello fornito dalla (e 4.6). Tuttavia, per $n \to \infty$, $t_{n-1,1-\frac{\alpha}{2}} \to z_{1-\frac{\alpha}{2}}$; in particolare $t_{40,0,95}$ differisce dalla $z_{0,95}$ per meno del 3% [10].

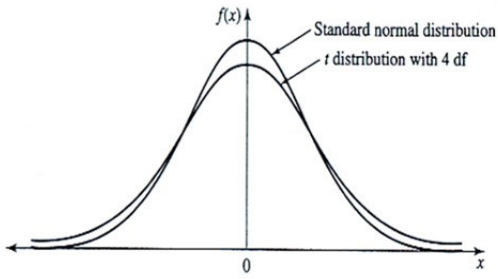



**Figura 4.1 Confronto tra la distribuzione normale standard e la distribuzione t**

Il metodo di analisi dell'output che utilizza la t di Student per il calcolo dell'intervallo di confidenza è definito *procedura con dimensione campionaria fissata.* Lo svantaggio di questa tecnica è che non consente all'analista di controllare la semi-ampiezza dell'intervallo di confidenza, ovvero la precisione della stima $\overline{X}(n)$. Infatti per n fissato, la semi-ampiezza dell'intervallo di confidenza dipende solo dalla varianza di X che non è nota ne controllabile [10].
L'intervallo di confidenza al $100 \cdot (1-\alpha)\%$ è per definizione:

*Stima* $\pm$ *margine di errore*

e rappresenta l'intervallo che contiene il valore vero del parametro della popolazione con una probabilità al più pari a $1-\alpha$. L'entità dell'errore si riduce all'aumentare della dimensione del campione mentre aumenta all'aumentare della deviazione standard della popolazione [86]. Non è possibile intervenire sulla varianza della popolazione per ridurre l'entità dell'errore ma è comunque possibile determinare la dimensione campionaria n che consente di ottenere l'errore desiderato [8].
Si introdurrà, in seguito, una procedura che consente di determinare il numero di repliche richieste per stimare la media $\mu = E(X)$ con un errore o precisione specificati.
Esistono due modi per misurare l'errore che si commette:

$$\textit{Errore assoluto } \beta = \left|\overline{X}(n) - \mu\right|$$

$$\textit{Errore relativo } \gamma = \frac{\left|\overline{X}(n) - \mu\right|}{\left|\mu\right|}$$

Se si effettuano n repliche di una simulazione fino a che la semi-ampiezza $\delta(n,\alpha)$ dell'intervallo di confidenza al $100 \cdot (1-\alpha)\%$ soddisfa la disuguaglianza

$\delta(n,\alpha) \le \beta$,

allora si ha:

$$1-\alpha \approx \Pr\left\{\overline{X}(n) - \delta(n,\alpha) \le \mu \le \overline{X}(n) + \delta(n,\alpha)\right\} = \Pr\left\{\left|\overline{X}(n) - \mu\right| \le \delta(n,\alpha)\right\} \le$$
$$\le \Pr\left\{\left|\overline{X}(n) - \mu\right| \le \beta\right\}$$

Quindi l'errore commesso è pari al più a $\beta$ con probabilità pari a circa $(1-\alpha)$. Se, ad esempio si costruiscono 100 intervalli di confidenza al 90% utilizzando il criterio di arresto sul numero delle repliche, ci si aspetta che l'errore assoluto $\left|\overline{X}(n) - \mu\right|$ sia al più pari a $\beta$ in circa in 90 dei 100 casi, mentre nei rimanenti 10 casi l'errore assoluto potrebbe essere maggiore.

Per quanto riguarda l'errore relativo, si supponga di eseguire n repliche di una simulazione fino a che sia verificata la seguente disuguaglianza:

$$\frac{\delta(n,\alpha)}{\left|\overline{X}(n)\right|} \le \gamma.$$

Analogamente al caso precedente si ha:

$$1-\alpha \approx \Pr\left\{\left|\overline{X}(n) - \mu\right| \le \delta(n,\alpha)\right\} = \Pr\left\{\frac{\left|\overline{X}(n) - \mu\right|}{\left|\overline{X}(n)\right|} \le \frac{\delta(n,\alpha)}{\left|\overline{X}(n)\right|}\right\} \le$$
$$\le \Pr\left\{\frac{\left|\overline{X}(n) - \mu\right|}{\left|\overline{X}(n)\right|} \le \gamma\right\} = \Pr\left\{\left|\overline{X}(n) - \mu\right| \le \gamma\left|\overline{X}(n) - \mu + \mu\right|\right\} \le$$
$$\le \Pr\left\{(1-\gamma)\left|\overline{X}(n) - \mu\right| \le \gamma|\mu|\right\} = \Pr\left\{\frac{\left|\overline{X}(n) - \mu\right|}{\left|\mu\right|} \le \frac{\gamma}{1-\gamma}\right\}$$

Quindi l'errore relativo è pari al più a $\gamma/(1-\gamma)$ con probabilità $1-\alpha$, ovvero se si costruiscono 100 intervalli di confidenza al 90%, indipendenti, utilizzando il criterio di arresto sul numero delle repliche adottato, ci si aspetta che l'errore relativo sia pari al più a $\gamma/(1-\gamma)$ in circa 90 dei 100 casi e nei rimanenti 10 potrà essere più grande di $\gamma/(1-\gamma)$ [10].

Se n fosse fissato a priori, non ci sarebbe nessuna possibilità di controllare la precisione della stima $\overline{X}(n)$, ovvero l'ampiezza dell'intervallo di confidenza. Quindi è necessario considerare la possibilità di decidere il numero di repliche da effettuare in modo da raggiungere una precisione desiderata. A questo scopo sono state definite due strategie generali:

- *procedura a due fasi:* in una prima fase vengono effettuate $n_0$ repliche sulla base delle quali si calcola $S^2(n_0)$ e la semi-ampiezza dell'intervallo di confidenza $\delta(n_0,\alpha)$. In seguito, eventualmente, si effettuano altre repliche fino al raggiungimento della precisione desiderata;
- *procedura iterativa:* si aumenta iterativamente il numero delle repliche fino a che non si raggiunge il livello di accuratezza desiderato.

***Procedura a due fasi***

Nella prima fase vengono effettuate un numero fissato $n_0$ di repliche e sulla base di queste si calcola la stima della varianza $S^2(n_0)$ e l'intervallo di confidenza per $\mu$. A questo punto, nella seconda fase, assumendo che la stima $S^2(n_0)$ non cambi significativamente all'aumentare del numero delle repliche, si effettuano eventualmente ulteriori repliche fino ad ottenere la precisione desiderata. Considerando l'*errore assoluto*, il numero totale delle repliche da effettuare per ottenere un errore assoluto pari al più a $\beta$ è dato da

$$n_a^* = \min\left\{ i \geq n \mid t_{i-1,1-\frac{\alpha}{2}} \sqrt{\frac{S^2(n_0)}{i}} \leq \beta \right\}$$

ovvero si incrementa il numero delle repliche aggiuntive $i$ di 1 fino a quando si verifica la disuguaglianza. Ovviamente, se $n_a^* > n_0$ allora si dovranno effettuare $n_a^* - n_0$ repliche aggiuntive.

Analogamente, considerando l'*errore relativo*, supponendo di aver costruito un intervallo di confidenza per $\mu$ basato su un numero fissato $n_0$ di repliche, allora si può verificare che il numero totale delle repliche da effettuare per ottenere un errore relativo $\gamma$ è

$$n_r^* = \min\left\{ i \geq n : \frac{t_{i-1,1-\frac{\alpha}{2}} \sqrt{\frac{S^2(n_0)}{i}}}{\left|\overline{X}(n_0)\right|} \leq \frac{\gamma}{1+\gamma} \right\}.$$

Anche in questo caso, se tale valore $n_r^*$ è maggiore del numero delle repliche già effettuate $n_0$, sarà necessario eseguire $n_r^* - n_0$ repliche aggiuntive fino al soddisfacimento della disuguaglianza.

Il termine $\gamma' = \gamma/1+\gamma$ è un aggiustamento che si rende necessario per ottenere l'errore relativo desiderato.

***Procedura iterativa***

La procedura a due fasi presenta l'inconveniente di utilizzare la stima $\overline{X}(n_0)$ e $S^2(n_0)$ basate sulle $n_0$ repliche fissate nel calcolo del numero di repliche necessarie per ottenere una precisione desiderata. Tuttavia tali stime potrebbero essere imprecise e questo potrebbe portare ad una scelta del numero di repliche troppo grande con notevole spreco di tempo di calcolo, oppure tale numero potrebbe essere troppo piccolo e quindi, di fatto, non si otterrebbe la precisione desiderata. Questo inconveniente può essere superato utilizzando una procedura iterativa che ha lo scopo di determinare una stima di $\mu$ con errore relativo pari a $\gamma$ e intervallo di confidenza del $100 \cdot (1-\alpha)\%$. La differenza fondamentale rispetto alla procedura a due fasi sta nel fatto che, ad ogni replica aggiuntiva eventualmente effettuata, viene ricalcolata la stima della varianza che, invece, nel caso precedente rimaneva fissata a $S^2(n_0)$.

Uno schema algoritmico di questa procedura è il seguente:

***Passo 1**: si effettuano $n_0$ repliche della simulazione e si pone $n = n_0$;*

***Passo 2:** si calcolano $\overline{X}(n)$ e $\delta(n,\alpha)$ da $X_1,...,X_n$;*

***Passo 3:** se $\frac{\delta(n,\alpha)}{\overline{X}(n)} \leq \frac{\gamma}{1+\gamma}$ si usa $\overline{X}(n)$ come stima di $\mu$* e STOP;

Da una sperimentazione diretta su un gran numero di modelli e di distribuzioni di probabilità per le quali sono noti i valori di $\mu$ è stato verificato che la procedura descritta fornisce una buona copertura dell'intervallo di confidenza al 90%, utilizzando $n_0 \geq 10$ e $\gamma \leq 0.15$.

Naturalmente la stessa procedura può essere utilizzata considerando l'errore assoluto al posto dell'errore relativo; tuttavia, il fatto che l'errore assoluto sia piccolo dipende dal modello adoperato e la procedura risulta molto sensibile alla scelta di $\beta$, di conseguenza si preferisce una procedura basata sull'errore relativo [10].

### *4.2.2. Stima di altre misure di prestazione*

Per comprendere il comportamento di un sistema manifatturiero è possibile stimare, oltre alle medie, altre misure di prestazione. Ad esempio, se si vuole stimare il numero atteso di elementi in coda in un determinato arco temporale, uno stimatore corretto è:

$$E\left[\frac{\int_0^T Q(t)dt}{T}\right]$$

dove $Q(t)$ è il numero di elementi in coda al tempo $t$ e $T$ è il periodo di attività del sistema (ad esempio un turno lavorativo).

Per la comprensione del sistema in esame, spesso può essere utile stimare la probabilità che una certa v.a. di interesse assuma dei valori compresi in un determinato intervallo. Sia X una variabile aleatoria definita su una replica di una simulazione e si supponga di voler stimare la probabilità:

$$\Pr\{X \in B\}$$

con $B$ insieme di numeri reali. Siano $X_1, X_2, ..., X_n$ le variabili aleatorie IID ottenute effettuando $n$ repliche indipendenti della simulazione. Se con $S$ si indica il numero delle $X_i$ che cade in $B$, $S$ avrà una distribuzione binomiale di parametri $n$ e $p$, e uno stimatore corretto di $p$ è dato da

$$\hat{p} = \frac{S}{n}.$$

Spesso può essere interessante anche ricavare una stima del $q$-quantile $x_q$ di una variabile aleatoria. Il $q$-quantile di una v.a. è quel numero tale che:

$$\Pr\{Y \le y_q\} = q.$$

In generale, se $X_{(1)}, X_{(2)}, ..., X_{(n)}$ sono le osservazioni ordinate della v.a. $X$, ottenute da n repliche indipendenti della simulazione, una stima di $x_q$ è fornita da:

$$\hat{x}_q = \begin{cases} X_{(nq)} & nq \quad \text{intero} \\ X_{(nq+1)} & \textit{altrimenti} \end{cases}$$

Gli intervalli di confidenza per queste stime si calcolano applicando le tecniche precedentemente analizzate [146].

## 4.3. Simulazione senza terminazione: analisi dello stato stazionario

Sia $Y_1, Y_2, ...$ un processo stocastico di output generato da un singolo run di una simulazione senza terminazione e si supponga che, per $i \to \infty$ sia:

$F_i(y) = \Pr\{Y_i \le y\} \to F(y) = \Pr\{Y \le y\}$

dove Y è la v.a. stazionaria di interesse con funzione di distribuzione $F$.

Con la notazione precedente è stata soppressa la dipendenza di $F_i$ dalle condizioni iniziali. Allora $\phi$ è un parametro stazionario se esso rappresenta una caratteristica di Y come, ad esempio il valore atteso $E(Y)$ o un quantile di Y, etc.

La difficoltà che si incontra nello stimare $\phi$ in una simulazione senza terminazione è che la funzione di distribuzione delle $Y_i$, $F_i$, è diversa da $F$ poiché dipendente dalle condizioni iniziali $I$, per cui le prime $l$ osservazioni $Y_1, Y_2, ..., Y_l$, per valori finiti di $l$, non possono essere considerate per il calcolo di $\phi$, in quanto si otterrebbe un indicatore non rappresentativo del comportamento stazionario del sistema. Ad esempio, la media campionaria $\overline{Y}(l)$ basata sulle prime $l$ osservazioni, con $l$ finito, risulterebbe uno stimatore distorto della media $\nu = E(Y)$.

Tale problema è noto in letteratura come *problema del transitorio iniziale* o *problema dello start-up* [10].

### *4.3.1. Il problema del transitorio iniziale*

I metodi più diffusi per determinare la lunghezza del warm-up, intesa come il periodo di simulazione prima di ottenere lo stato stazionario, sono [113]:

- ***METODO DI WELCH;***
- ***METODO SPC;***
- ***RANDOMIZATION TEST;***
- ***CONWAY RULE;***
- ***CROSSING OF THE MEANS RULE;***
- ***MARGINAL STANDARD ERROR RULE.***

Si supponga di voler stimare la media stazionaria, definita come

$$\nu = \lim_{i \to \infty} E(Y_i) = E(Y)$$

Per porre rimedio agli eventuali problemi che si possono verificare in fase di startup si ricorre alla *tecnica di cancellazione dei dati iniziali,* detta anche *warming up* del modello, che consiste nel non considerare nella stima le prime osservazioni che sono quelle più influenzate dalle condizioni iniziali.


**Figura 4.2 Il periodo di warming-up**

Quindi, invece di utilizzare la stima $\overline{Y}(m)$ basata su m osservazioni, si considera:

$$\overline{Y}(m,l) = \frac{\sum_{j=l+1}^{m} Y_j}{m-l}$$

dove $l$ è il numero finito delle osservazioni che vengono scartate.

Ovviamente si crea il problema relativo alla corretta scelta di $l$, ovvero del periodo di warming up, di modo che:

$$E(\overline{Y}(m,l)) \approx \nu$$

Se il valore di $l$ è troppo piccolo c'è il rischio che la stima risenta delle condizioni iniziali, mentre se è troppo grande si potrebbe avare uno spreco di tempo di calcolo.

È stata proposta una procedura per determinare $l$ basata su un'analisi grafica: poiché si vuole determinare un valore di $l$ tale che, per $i > l$, risulti $E(Y_i) \approx \nu$, graficamente questo si traduce nel determinare quando la curva $E(Y_i), i = 1,2,\ldots$, si stabilizza intorno al valore $\nu$. Il problema è che spesso si ha un andamento influenzato dall'alta variabilità del processo $Y_1, Y_2, \ldots$. Per superare questo inconveniente, si introduce la *procedura di Welch* che, prima di effettuare l'analisi grafica, prevede un trattamento dei dati per ridurre la varianza. Uno schema della procedura di Welch è il seguente:

**Passo 1:** Si effettuano n repliche ciascuna di lunghezza m, con m grande. Sia $Y_{ji}$ la $i-esima$ osservazione della $j-esima$ replica ($j=1,2,...,n; i=1,2,...,m$):

**Passo 2:** Si calcola la media $\overline{Y}_i = \sum_{j=1}^{n} \frac{Y_{ji}}{n}$, per $i=1,2,...,m$, tra i dati corrispondenti nelle diverse repliche. Il processo stocastico delle medie $\overline{Y}_1, \overline{Y}_2, ...$ ha media $E(\overline{Y}_i) = E(Y_i)$ e varianza $Var(\overline{Y}_i) = Var(Y_i)/n$, dunque la media è rimasta invariata, mentre la varianza è stata ridotta di un fattore pari a $1/n$.

**Passo 3:** Si sostituisce ciascun termine del processo stocastico delle medie $\overline{Y}_1, \overline{Y}_2, ...$ con la media mobile $\overline{Y}_i(w)$, definita come

$$\overline{Y}_i(w) = \begin{cases} \dfrac{\sum_{s=-w}^{w} \overline{Y}_{i+s}}{2w+1} & i = w+1,...,m-w \\ \dfrac{\sum_{s=-(i-1)}^{i-1} \overline{Y}_{i+s}}{2i-1} & i = 1,...,w \end{cases}$$

*dove w è un intero positivo tale che* $w \le m/4$*, definito "time window". La media mobile* $\overline{Y}_i(w)$ *non è altro che la media campionaria di* $2w+1$ *osservazioni delle medie* $\overline{Y}_i$ *centrate sull'i- esima osservazione.*

***Passo 4:*** *Si disegna il grafico delle* $\overline{Y}_i(w)$*, per* $i=1,2,...,m-w$*, e si sceglie il valore di l oltre il quale la successione* $\{\overline{Y}_i(w)\}$ *appare giunta a convergenza.*

È molto importante scegliere in maniera appropriata i tre parametri della simulazione, *m, n* e *w*:

- la lunghezza delle repliche *m* dovrà essere più grande del valore che ci si aspetta per *l* e tale da permettere nella simulazione un numero elevato di occorrenze di tutti gli eventi, anche quelli poco probabili;

- per quanto riguarda la scelta del numero di repliche *n*, è opportuno iniziare con valori di *n* pari a 5 o a 10 per poi aumentare, se necessario;
- il valore del time window $w$ deve essere sufficientemente grande da rendere regolare il grafico delle $\overline{Y}_i(w)$, ma non tale da non permettere l'individuazione del transitorio [1, 24, 25, 34].

Anche nel metodo SPC, ampiamente illustrato da Robinson, è necessario effettuare *n* replicazioni e calcolare la media per ciascuna di queste. L'insieme delle medie $\{\overline{Y}_i : i=1,2,\ldots,m\}$ è suddivise in *b* batch di dimensione *k*, la cui media è definita come $\overline{\overline{Y}}_x$ con x= 1,2, …,b. La dimensione *b* è scelta utilizzando test di correlazione di Anderson Darling e quello di normalità di Von Neumann, tuttavia, in genere, non si usano meno di 20 batch. Si ottiene in questo modo una serie in funzione del tempo

$Y_{(k)}= \{ \overline{\overline{Y}}_1(k),\ldots,\overline{\overline{Y}}_b(k)\}$. Si procede alla stima della media e della deviazione standard per l'ultima metà della serie e si definiscono i tre limiti di controllo impiegati in una carta di controllo in cui sono rappresentate la serie temporale e la media stimata. I limiti di controllo sono così calcolati:

$$CL = \hat{\mu} \pm z\hat{\sigma} / \sqrt{b/2} \text{ con } z = 1,2,3$$

Lo stato stazionario è raggiunto quando il processo è e resta in controllo. Il caso di processo fuori controllo è evidenziato nei seguenti casi:

1. un punto è fuori dai limiti di controllo a 3-sigma della carta;
2. due o tre punti consecutivi sono fuori dal limite di controllo 2-sigma;
3. quattro, cinque punti consecutivi sono fuori dal limite di controllo a 1-sigma;
4. otto punti consecutivi sono su un lato della media;
5. i punti iniziali sono tutti su un lato della media.

Il Randomization test, sviluppato da Yücesan, è utilizzato nella valutazione della media del processo. Il metodo si basa sull'ipotesi nulla che non ci sia bias di inizializzazione. Si procede ad effettuare una simulazione di lunghezza *m* ore e ottenendo la serie in output $\{Y_1,Y_2,\ldots,Y_m\}$. I dati sono raccolti in *b* batches di lunghezza *k*, per ognuno dei quali si calcola la media. Le medie sono raggruppate in due gruppi, ad esempio per la prima iterazione della procedura si considera il primo gruppo costituito dalla media del primo batch e il secondo dalle restanti *b*-

*1* medie. Si procede, quindi, al confronto tra le medie totali dei due gruppi. Se la differenza delle due medie è significativamente diversa da zero, allora l'ipotesi nulla può essere rigettata. In tal caso, si considerano come output della simulazione nello stato stazionario i dati contenuti nel secondo gruppo. Altrimenti, si ridefiniscono i due gruppi aggiungendo il secondo batch al primo gruppo.

Conway ha suggerito la seguente regola di troncamento: si troncano una serie di misure finchè il primo della serie non è nè il massimo nè il minimo della serie rimanente. Sulla base di tale regola è stato sviluppato il seguente algoritmo. Si effettuano *n* replicazioni di tentativo di lunghezza *m* e si calcolano $y^{+}_{jr}$ e $y^{-}_{jr}$ usando le seguenti relazioni:

$$y^{+}_{jr}=\max(y_{jl} : l=r,\ldots,m) \text{ con } j=1,\ldots,n$$
$$y^{-}_{jr}=\min(y_{jl} : l=r,\ldots,m) \text{ con } j=1,\ldots,n$$

e si determina per r=1,2,…,m il $t_j$ tale che $t_j = \min_r \{ y^{-}_{jr} < y_{jr} < y^{+}_{jr}\}$ si verifica per il rimo periodo e si stima il punto di troncamento $t^{*}= \max\{t_1,t_2,\ldots,t_n\}$.

Nel metodo Crossing of the means rule si generano gli output $\{Y_1,Y_2,\ldots,Y_m\}$ e si definisce $w_j$, data $\overline{Y}_m = \sum_{j=1}^{m} \frac{Y_j}{n}$, come:

$$w_j = \begin{cases} 1 & se Y_j > \overline{Y}_m, \quad Y_{j+1} < \overline{Y}_m \\ & oppure Y_j < \overline{Y}_m \quad Y_{j+1} > \overline{Y}_m \\ 0 & altrimenti \end{cases}$$

con j=1,2,….,m-1

e si calcola il numero di volte in cui la serie attraversa la media come:

$$\Omega_m = \sum_{j=1}^{m-1} w_j \text{ con m=1,2,…,1}$$

tale che ad *l* il numero di attraversamenti è uguale ad un numero predefinito.

L'ultimo metodo consiste nel definire *m* come la dimensione del batch, *b* il numero di batch e *n* la lunghezza della simulazione. Secondo la regola MSER, per un processo stocastico finito, $\{Y_i(j): i=1,2,\ldots,n\}$ il punto di troncatura ottimale è dato da:

$$d_j^* = \min_{n>d(j)\geq 0}\left[\frac{z_{\alpha/2}s(d(j))}{\sqrt{n(j)-d(j)}}\right]$$

Quindi, attraverso la soluzione di un problema non vincolato di minimizzazione si ottiene l'istante di troncatura.

A questo punto, dette, per esempio, $l$ il numero di osservazioni troncate dalle $n$ totali con uno dei metodi esposti la media sarà calcolata come:

$$\overline{Y}_{n,l} = \frac{1}{n-l}\sum_{i=l+1}^{n} Y_i$$

### *4.3.2. Stima della media stazionaria e intervalli di confidenza*

Si analizzeranno i diversi metodi proposti in letteratura per la stima della media stazionaria $\nu$ del processo di output $Y = \{Y_i, i = 1,2,...,n\}$, definita come:

$$\nu = \lim_{i\to\infty} E(Y_i) = E(Y)$$

Stima della media basata sull'approccio "repliche/cancellazioni"

Questo metodo fornisce una stima della media stazionaria $\nu$ utilizzando, in ciascuna replica, solo le osservazioni successive al periodo di warm up.

Si supponga di aver effettuato $n$ repliche di una simulazione, ciascuna di lunghezza $m$ (repliche pilota) e di aver determinato il valore della lunghezza del transitorio, $l$. Sia $Y_{ji}$ la $i$-esima osservazione prodotta nella $j$-esima replica e sia $X_j$ una nuova v.a. definita come

$$X_j = \frac{\sum_{i=l+1}^{m} Y_{ji}}{m-l} \qquad \text{per } j = 1,...,n$$

che rappresenta la media campionaria delle osservazioni $Y_{j,l+1}, Y_{j,l+2},..., Y_{j,m}$ prodotte dalla $j$-esima replica. Bisogna notare che le $Y_{ji}$ sono v.a. dipendenti perchè prodotte dalla stessa replica, mentre le $X_j$ sono v.a. indipendenti ed identicamente distribuite perché prodotte da repliche indipendenti. Inoltre, se $m$ è sufficientemente grande, risulta

$$\nu = E(X_j).$$

Al termine dell' *n*-esima replica si calcola la media campionaria delle $X_j$

$$\overline{X}(n) = \frac{1}{n}\sum_{j=1}^{n} X_j$$

e la varianza campionaria delle $X_j$, che può essere calcolata con la formula

$$S^2(n) = \frac{1}{n-1}\sum_{j=1}^{n}(X_j - \overline{X}(n))^2$$

essendo le $X_j$ v.a. I.I.D.

Dunque l'intervallo di confidenza per $\nu$ al $100\cdot(1-\alpha)\%$ è dato da

$$\left[\overline{X}(n) - t_{n-1,1-\frac{\alpha}{2}}\frac{S(n)}{\sqrt{n}}, \overline{X}(n) + t_{n-1,1-\frac{\alpha}{2}}\frac{S(n)}{\sqrt{n}}\right].$$

Nella maggior parte dei casi, per il basso costo del tempo di calcolo, non costituisce un problema il ricorso ad *n* repliche pilota per determinare *l* e poi l'uso delle sole ultime *m-l* osservazioni provenienti da *n'* repliche differenti da quelle pilota. È comunque possibile utilizzare un unico set di *n* repliche sia per il calcolo del periodo di warm up, sia per la costruzione dell'intervallo di confidenza, se la lunghezza *m* di ciascuna replica, inizialmente fissata, è molto più grande di *l.*
Continuano a valere tutte le considerazioni fatte circa la possibilità di ottenere una precisione desiderata [10].

<u>Metodo basato sull'evoluzione della MSPE</u>

La determinazione della lunghezza del run di simulazione è uno dei punti critici dell'esperimento di simulazione. Le decisioni all'interno di una gestione tradizionale di un esperimento fanno riferimento ad un compromesso tra l'affidabilità dei risultati cercati e il costo e/o tempo di sperimentazione. Entrambi si incrementano all'aumentare della complessità del modello. La critica fatta da Mosca et al. [122] delle metodologie esistenti volge su tre punti fondamentali:

- non sono capaci di fornire una stima a priori sull'entità del puro errore sperimentale che affligge gli output
- richiedono un rilevante numero di lanci addizionali che non possono venire riutilizzati nella stima degli output, facendone derivare un non gradito incremento dei tempi di sperimentazione e infine

- forniscono risultati paragonabili a quelli stimabili ad intuito da un ricercatore sufficientemente abile e pratico del simulatore.

Caratteristiche della metodologia ideale sono quelle di non richiedere prove aggiuntive non recuperabili, di consentire una migliore ripartizione del tempo totale di sperimentazione, ossia diminuendo la durata di ogni singolo run, posso effettuare, a parità di tempo di sperimentazione, un numero più elevato di lanci e infine consente di misurare il puro errore sperimentale e la sua evoluzione temporale sull'arco di tempo simulato.

Volendo portare avanti una strategia di esperimenti progettati (p.c.c.), è necessario calcolare la durata ottima di ogni singolo run di simulazione al fine di ridurre costi e tempi di sperimentazione. L'idea base della metodologia corretta di misura della durata si basa sui presupposti che le $n_0$ prove centrali di un p.c.c. consentono una misura dell'errore sperimentale (puro errore) e nel caso si assenza di un p.c.c., la metodologia può essere ugualmente implementata replicando l'esperimento, in uno stesso punto di progetto, per 3÷5 volte (accertandosi che ad ogni nuovo lancio corrispondano diversi semi di innesco dei generatori dei numeri random). Altro presupposto è quello che essendo in accordo con il teorema di Cochran, l'Errore Sperimentale è distribuito secondo una distribuzione normale di media 0 e varianza $\sigma^2$:

$$[NID\ (0, \sigma^2)]$$

Per cui il valore atteso della Mean Square dell'errore coincide con $\sigma^2$:

$$E\ (MS_E) = \sigma^2$$

Inoltre, l'errore sperimentale, nella simulazione Montecarlo, decresce statisticamente fino al raggiungimento del valore limite stabile al crescere della lunghezza del lancio in quanto tanto maggiore è il numero di estrazioni effettuate delle variabili casuali presenti nel modello, tanto più ogni distribuzione viene meglio "ricoperta" dai valori campionati. Partendo da queste considerazioni appena effettuate, si suppose che il comportamento dell'errore sperimentale, stimato attraverso la Mean Square Pure Error, avesse un andamento del tipo "Knee Curve", ossia curva a ginocchio:

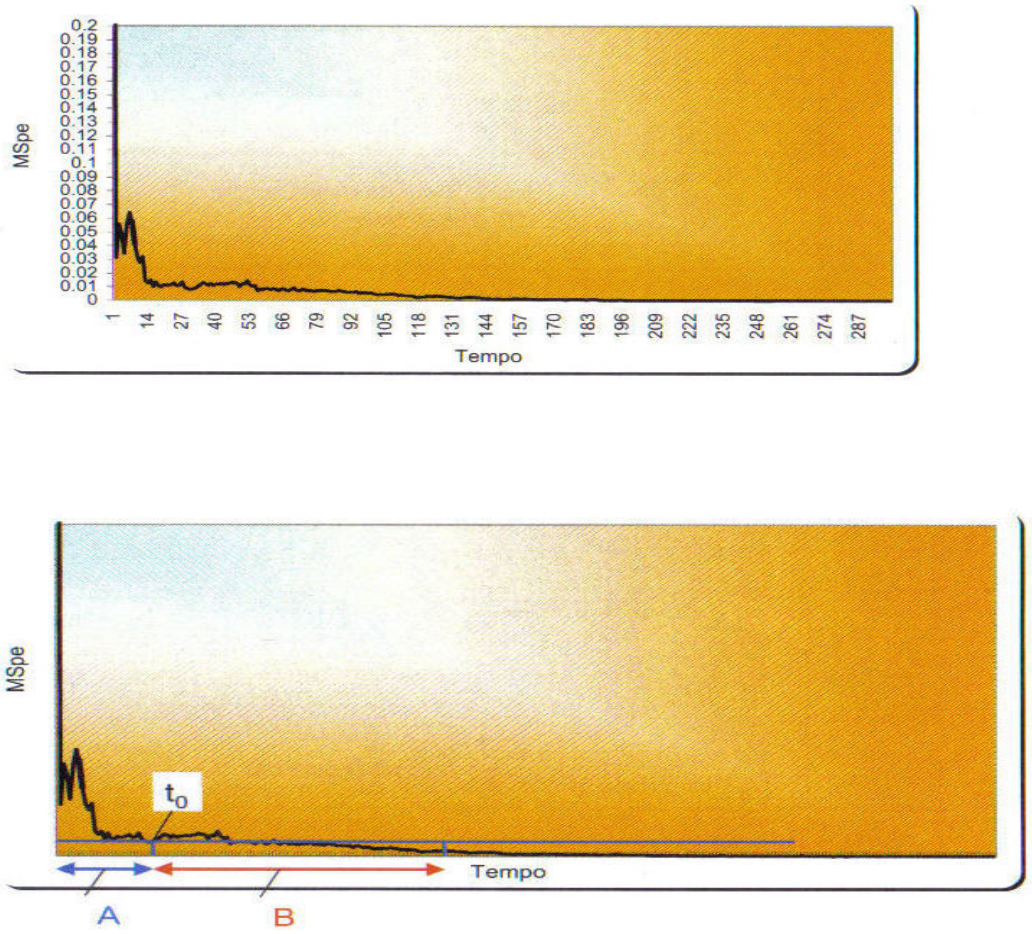



**Figura 4.3 Esempio di Knee curve**

Supponendo quindi che sia così, esiste sempre un istante $t_0$ oltre il quale non si ha interesse a protrarre l'esperimento in quanto a grandi incrementi del tempo T, corrispondono decrementi poco significativi della Mspe. Volendo impostare i passi fondamentali della metodologia, dato un modello testato la sua convalida statistica prevede:

1. Stima di un tempo di simulazione T' tale da "entrare " nella zona di stabilizzazione della Mspe (zona B), oltre il ginocchio della curva ( T' >> $t_0$);
2. Stima del tempo di inizializzazione con contabilizzazione dei risultati solo dopo lo stesso (che tuttavia può anche non essere considerata in quanto il suo effetto sull'Mspe a regime è ininfluente);

3. Lancio delle $n_0$ prove centrali (ovvero delle prove replicate) con effettuazione delle r rilevazioni intermedie del valore delle funzioni obiettivo considerate ad intervalli di tempo $t_i$ (i=1,…,r) equispaziati. Ciò equivale al lancio di r simulazioni di durata $t_i$, ciascuna replicata $n_0$ volte;
4. Stima dell'errore sperimentale (Mspe) per ciascuno degli $n_0$ set di lanci aventi una lunghezza $t_i$ omogenea attraverso le relazioni note:

$$SSpe(t_i) = \sum_{j=1}^{n_0} \left(Y_j(t_i) - \overline{Y}(t_i)\right)^2$$

$$MSpe(t_i) = \frac{SSpe(t_i)}{n_0 - 1}$$

5. Analisi della evoluzione temporale della Mspe, ottenuta tramite il procedimento appena descritto (una per ogni valore temporale istantaneo: 1000,2000,…,10000). Si consideri che quanto maggiore risulta la durata di una simulazione tanto meglio i valori estratti approssimano le distribuzioni di frequenza originarie: le risposte tendono quindi ad addensarsi attorno al loro valore medio. Per questo motivo al crescere di $t_i$ ci si attende una progressiva diminuzione ella Mspe fino al raggiungimento della stabilizzazione provocata da una crescente insensibilità del sistema nei confronti di nuove estrazioni casuali: nel periodo iniziale ogni campionamento determina vistose oscillazioni nelle medie della risposta, mentre al trascorrere del tempo l'influenza del numero random sulla stima di quest'ultima perde via via di significatività;
6. Costruzione del grafico della durata portando in ascissa il tempo simulato e in ordinata i relativi valori della Mspe agli istanti $t_{i;}$
7. scelta dell'istante di fine simulazione attraverso una analisi visiva del diagramma (Mspe,$t_i$).

Inoltre è possibile, onde evitare errori interpretativi dovuti all'impossibilità di utilizzare scale grafiche adeguate, utilizzare una procedura più rigorosa sotto il profilo strettamente statistico:
si suddivide l'asse dei tempi in un numero opportuno di classi di ampiezza sufficientemente significativa e si calcola nell'ambito di ogni classe per ciascun obiettivo, la media d e le somme dei quadrati SSD delle SSpe.
Detto t il numero di osservazioni per ogni classe, si otterrà per ciascuna classe:

$$\overline{D} = \sum_{i=1}^{t} \frac{SSpe_i}{t} \qquad SSD = \sum_{i=1}^{t} \left(SSpe_i - \overline{D}\right)^2$$

Le rispettive medie quadratiche MSD si ottengono allora dividendo le somme dei quadrati per il numero di gradi di libertà relativo ad ogni blocco e dato da (t-1). Quindi:

$$MSD = \frac{SSD}{(t-1)}$$

Si confrontano a questo punto le medie quadratiche di ogni classe con quelle ad esse immediatamente successive, rapportando sempre quella di maggiore entità a quella meno rilevante, in modo da ottenere frazioni comunque maggiori dell'unità. Così facendo, poiché i rapporti tra medie di quadrati costituiscono grandezze distribuite secondo una Fisher, si perviene ad un test statistico che consente di apprezzare la significatività delle variazioni nel tempo delle MSpe.

Le due ipotesi da confrontare in tale test sono

$H_0$: (ipotesi nulla) mancanza di significatività tra la media dei quadrati MSD di una classe e quella della successiva.

$H_1$: (ipotesi alternativa) variazione significativa tra le due classi.

Scelto un valore $\alpha$ relativo all'errore di I specie e indicato con k il numero totale delle classi si provvede a calcolare i rapporti:

$$F_j = \frac{MSD_j}{MSD_{j+1}} \qquad \text{oppure} \qquad F_j = \frac{MSD_{j+1}}{MSD_j}$$

Quindi si può concludere che da un certo istante in poi le MSpe si sono significativamente assestate se:

$$F_j < F_{\alpha,t-1,t-1}$$

Se la grandezza $F_j$, e nessun altra grandezza $F_i$ con i>j, rispetto a tale relazione soddisfa l'ipotesi alternativa, si può senz'altro fissare il centro della esima classe come istante di fine simulazione.

Volendo effettuare una analisi della metodologia si può notare come nella simulazione la durata ottima è indipendente dalla scelta dei numeri random. La figura che segue conferma questa affermazione: è stata ricavata attraverso il lancio di 4 distinti set di prove $n_0$ con identiche condizioni al contorno e differenziazione dei soli semi di innesco dei set di numeri random utilizzati:

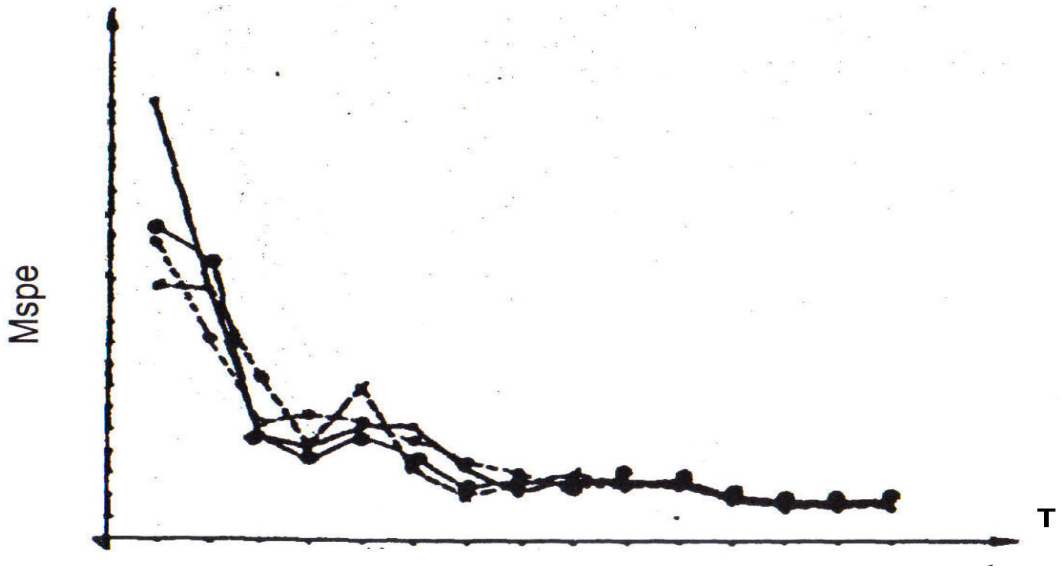



**Figura 4.4 Durata ottima indipendente dalla scelta dei numeri random: 4 distinti set di prove**

Inoltre anche l'autocorrelazione delle risposte è ininfluente. Il problema prospettato nasce dal fatto che la storia di ogni sistema simulato ad un dato istanti $t_i$ è influenzata dagli istanti precedenti ( $t_0$ a $t_{i-1}$) e condiziona quelli degli istanti successivi (da $t_{i+1}$ a $t_r$ ). La procedura presuppone per contro, che le osservazioni in ciascuna delle $r*n_0$ simulazioni fittizie siano invece indipendenti in quanto l'eventuale correlazione tra i blocchi inficerebbe tutti i risultati. Concettualmente questo tipo di rischio viene aggirato effettuando le rilevazioni intermedie ad intervalli di tempo opportunamente distanziati: in tal modo solo i primi "pochi" eventi del sub-run i-esimo risultano correlati con gli ultimi del sub-run i-1 per cui la media $y(t_i)$ e $y(t_{i-1})$ rivelano correlazioni di scarsa entità. Lavorando, in definitiva, attraverso un "taglio" orizzontale delle $n_0$ simulazioni agli istanti di tempo prescelti (1000÷10000), si vengono a smorzare, fino ad annullare, gli effetti della correlazione esistente tra gli istanti successivi di una stessa prova di simulazione.

La Mspe stimata su funzioni obiettivo additive nel tempo restituisce ancora una curva di durata a ginocchio previa normalizzazione della Mspe. La normalizzazione del valore della risposta osservata avviene come segue:

$$Y = \frac{Y(t_i)}{t_i}$$

quindi si calcola la Mspe ai nuovi istanti con questa nuova valorizzazione.

Nel caso di una simulazione multiobiettivo, per la determinazione della durata ottima, l'approccio al problema è quello dettato dalla logica, ossia agendo su di un unico set di prove centrali costruisco tante tabelle di uscita quanti sono gli obiettivi considerati e calcolo le Mspe per ogni obiettivo. Determinato ciò, costruisco i grafici di evoluzione temporale delle Mspe per ogni obiettivo e infine determinati i tempi $T_1,\ldots,T_k$ di durata ottima singola, si scegli come tempo ottimale di run per l'intero esperimento, l'istante $T_x$ relativo alla curva che va a stabilizzazione nel maggior tempo.

Per quanto concerne la capacità del simulatore di essere aderente alla realtà in esame, essa va testata nel modo tradizionale e cioè confrontando i dati ottenuti tramite il simulatore con i dati di ritorno dalla realtà in esame o da realtà analoghe. La curva evolutiva della Mspe nel tempo ci permette di vagliare *l'affidabilità statistica del simulatore* in termini di regolarità della knee-curve e di entità della Mspe a regime (rumore di fondo). Si noti che in generale, un simulatore può essere idoneo alla simulazione di certe funzioni obiettivo sotto determinate condizioni al contorno e non di altre per cui, in un'ottica di confronto, occorrerà entrare nel maggior dettaglio possibile del modello. Tornando, invece, al rumore di fondo, questo non è una entità caratteristica di un certo simulatore ma dipende, nell'ambito del simulatore utilizzato da:

- livello di stocasticità assegnato alle variabili in casuali;
- la capacità del simulatore di trattare la stocasticità;
- tipo di funzione obiettivo analizzata

Per il primo punto, in quanto l'errore sperimentale al tempo $t_i$ misura la capacità del simulatore a fornire risposte tra loro congruenti, in termini di funzioni obiettivo, al variare dei soli di semi di innesco dei numeri random, una elevata stocasticità delle variabili casuali comporta una maggiore probabilità di ottenere, ad ogni lancio all'istante $t_i$, valori della variabile dipendente tra loro differenti (valori elevati di SSpe):

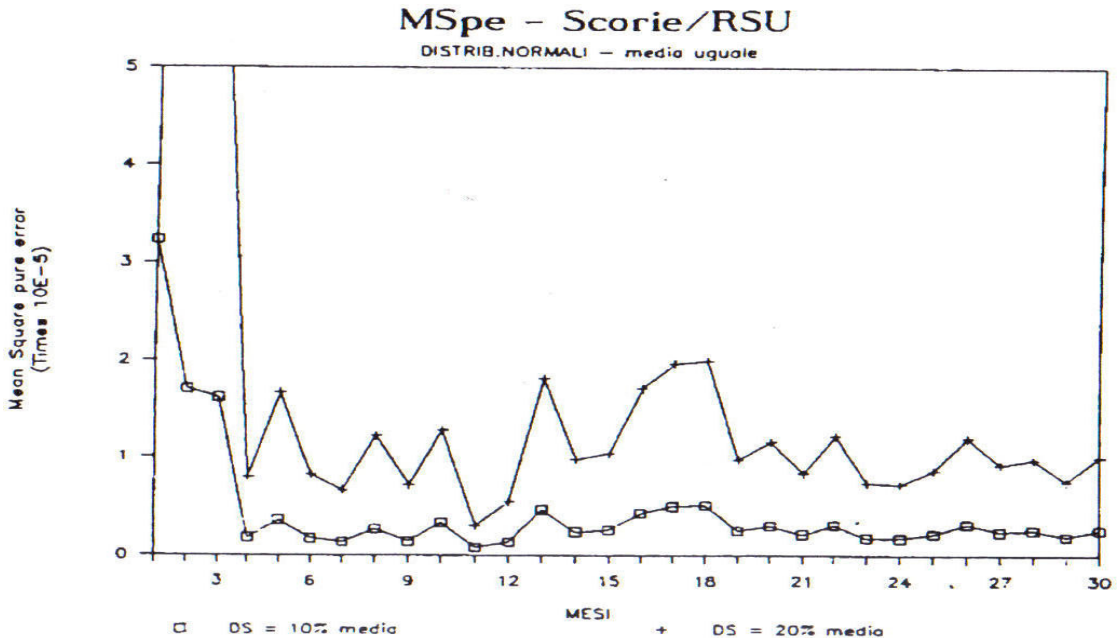



**Figura 4.5 Livello di stocasticità assegnato alle variabili casuali**

Il secondo punto si fonda sul diverso comportamento, a stabilizzazione, di simulatori ad eventi costruiti con tecniche differenti. In alcuni casi il simulatore dà un rumore di fondo accettabile solo entro certi livelli ben precisi di stocasticità. L'ultimo punto, invece, dipende dalla cura con cui sono state costruite le porzioni di programma e da come esse vengono sollecitate, nella simulazione, dalle diverse funzioni obiettivo.

Poiché l'errore sperimentale rappresenta sostanzialmente "la capacità del simulatore di fornire risposte congruenti, in presenza di uno stesso tipo di sollecitazione", si possono fare alcune considerazioni: una minima variabilità della Mspe, al variare della variabile dipendente considerata, indica una notevole costanza di reazione del modello ad un input (invariato) che sollecita, prevalentemente, certe zone del simulatore. Inoltre un elevato livello di incongruenza tra le risposte, indica che in fase di elaborazione, vengono sollecitate anche porzioni di programma sviluppate secondo canoni che risultano di minore affidabilità statistica.

Una osservazione fondamentale da effettuare è che essendo $E(MSpe) = \sigma^2$ il miglior estimatore della varianza dell'errore sperimentale supposto distribuito come una NID $(0, \sigma^2)$, l'entità del rumore di fondo deve essere strettamente correlata con l'entità numerica della funzione obiettivo che si sta indagando. Il motivo è quello di non correre il rischio che il valore (medio) della variabile

dipendente determinato, venga completamente mascherato dall'entità dello scarto quadratico medio, rappresentato appunto dal puro errore sperimentale. Per esempio una funzione obiettivo dell'ordine del $10^{-1} \div 10^{-2}$ (caso dei coefficienti di utilizzazione), dovrà avere un errore sperimentale a stabilizzazione, perlomeno dell'ordine $10^{-4}$.

Metodi basati su una singola replica

In letteratura sono stati proposti cinque metodi basati su una singola replica della stessa simulazione, di lunghezza fissata ad arbitrio. Nell'implementazione di tali tecniche, dato che le osservazioni vengono generate dalla stessa replica, e quindi non sono indipendenti, non è possibile calcolare l'intervallo di confidenza nel modo descritto, perché la varianza campionaria

$$S^2(n) = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X}(n))^2$$

è uno stimatore corretto della varianza $\sigma^2$ solo se le v.a. sono I.I.D.
Ciascun metodo propone una soluzione diversa a questo problema.

*Il metodo delle batch means*

Nell'ipotesi di avere ottenuto dei dati identicamente distribuiti e indipendenti, il metodo batching permette di impiegare un'unica simulazione suddivisa in batches per la stima dei parametri di interesse. Per determinare la lunghezza ottimale della simulazione, è necessario considerare le diverse tecniche connesse ai differenti metodi di valutazione degli stimatori dei parametri impiegati. Infatti, laddove si definisca uno stimatore di un parametro incognito, è necessario analizzare l'errore di campionamento valutato attraverso l'errore standard. Il primo errore si referisce al grado di precisione con cui lo stimatore è rappresentativo del valore vero. L'unica fonte di tale errore è il generatore di numeri casuali e il seme impiegato. Fondamentalmente, i metodi di valutazione dell'errore si basano sulla definizione della stima della varianza dello stimatore o della varianza asintotica, entrambe correlate alla lunghezza della simulazione e alle dimensioni dei batch. In ogni caso, si considera quale stimatore quello della media $\mu$ della popolazione di partenza, $\overline{X}_n$ tale che $E[\overline{X}_n] = \mu$. Si considerino *k*

batches ciascuno costituito da *b* osservazioni, con *n=kb* osservazioni totali. Per l'*i*-esimo batch si ottengono le osservazioni $\{Y_{(i-1)b+1}, Y_{(i-1)b+2},\ldots\ldots, Y_{ib}\}$ e la media per tale batch è:

$$\overline{Y}_i(b) = \frac{1}{b}\sum_{j=1}^{b} Y_{(i-1)b+j}$$

Definita $\sigma^2_n$=VAR($\overline{X}_n$), tale che $\lim_{n\to\infty} n * VAR(\overline{X}_n) = \sigma^2_\infty < \infty$, si ha che:

$$\sigma_n^2 = \frac{\sigma_b^2}{k}\left(1 + \frac{n\sigma_n^2 - b\sigma_b^2}{b\sigma_b^2}\right),$$

con $\sigma^2_b$ varianza della popolazione di partenza. In altri termini, $\sigma^2_b/k$ approssima $\sigma^2_n$ con un errore che diminuisce al tendere prima di *n* e poi di *b* all'infinito e di *n/b* a zero. Allo stesso modo, la correlazione tra le medie dei batches diminuisce. A questo punto, è possibile stimare la media totale e la varianza $\sigma^2_b$ come:

$$\overline{X}_n = \frac{1}{k}\sum_{i=1}^{k} \overline{Y}_i(b),$$

$$S^2(n,k) = \frac{1}{k-1}\sum_{i=1}^{k} (\overline{Y}_i(b) - \overline{X}_n)^2 ,$$

da cui è possibile calcolare l'intervallo di confidenza per la media μ. Infatti, nell'ipotesi di validità del teorema del limite centrale, $\overline{X}_n$ tende ad una gaussiana di media μ e varianza $\sigma^2_b/k$. In tal modo, essendo ignota la varianza è possibile utilizzare la funzione ancillare t-student con *k-1*gradi di libertà. L'intervallo di confidenza è dato, quindi, da:

$$\overline{X}_n \pm t_{k-1,1-\alpha/2}\sqrt{S^2(n,k)/k} \ .$$

In teoria, è possibile considerare gli stimatori dei parametri in output come suddivisi in due categorie, ovvero *Non overlapping Batch means estimetor* (NBM) e *Overlapping batch means estimetor* (OBM). In relazione alla tipologia di stimatori impiegati, è possibile determinare secondo diverse procedure, il numero ottimale di batches o la loro dimensione o entrambi i parametri. Per quanto concerne la prima tipologia di stimatori, per determinare le dimensioni dei batch e il numero sono state sviluppate differenti regole sulla base dell'ipotesi di validità del teorema del limite centrale, per il quale:

$$\sqrt{n}(\overline{X}_n - \mu) \xrightarrow{d} \sigma_\infty N(0,1) \text{ per } n\to\infty$$

Tali regole possono essere così esemplificate [9]:

- ***Fixed number of Batches (FNB);***
- ***Square root (SQRT);***
- ***LABATCH;***
- ***ABATCH.***

La regola FNB si basa sulla relazione tra $\sigma^2_b$ e $\sigma^2_n$, per la quale, al tendere all'infinito di *n*, $\sigma^2_b/k$ tende a $\sigma^2_n$. Per questo motivo, il numero di batches *k* è fissato e la dimensione varia con *n* secondo la relazione $b_n = \lfloor n/k \rfloor$. La limitazione di tale regola è che $b_n S^2(b)$ non è uno stimatore consistente di $\sigma^2_\infty$ e, quindi, l'intervallo di confidenza risulta più ampio rispetto a quello che sarebbe prodotto da uno stimatore consistente. Per questo motivo è stata introdotta la SQRT che modifica sia la dimensione dei batches che il numero, nell'ipotesi di validità dell'assunzione di approssimazione forte (ASA), per la quale esiste una costante $\lambda \in (0, ½]$ e una variabile casuale finita C tale che:

$$P\{|n(\overline{X}_n-\mu)-\sigma_\infty W(n)| \le Cn^{1/2-\lambda}\}=1 \text{ con } n\to\infty;$$

dove $\{W(n), t\ge 0\}$ è il processo di moto Browniano standard. Inoltre, λ è prossima a ½ per processi con ridotta autocorrelazione. La SQRT usa $b_n = \lfloor n^\theta \rfloor$ con $\theta \in (1-2\lambda, 1)$ . Da cui, se $n\to\infty$ si ha che:

$$\overline{X}_n \to \mu \text{ e } b_n S^2(b) \to \sigma^2_\infty \text{ e } Z_{k_n} = \frac{\overline{X}_n - \mu}{\sqrt{S^2(n,k_n)/k_n}} \xrightarrow{d} N(0,1).$$

L'impiego del pedice *n* per *k* è impiegato per sottolinearne la variabilità, mentre nel caso precedente *k* è fisso. In questo caso, l'intervallo di confidenza trovato risulta asintoticamente valido per μ. In particolare, θ è uguale a ½ per $¼ < \lambda < ½$ [7]. Tuttavia, è dimostrato che tale metodo sottostima la varianza della media campionaria. Per questo motivo, sono state sviluppate da Fisherman e Yarberry due regole iterative la LBATCH e la ABATCH . Entrambe effettuano il test di correlazione di Von Neumann ad ogni istante $n_l \approx n_l 2^{l-1}$ con i=1,2,… Il test consiste nella valutazione della statistica:

$$C(n,k) = \sqrt{\frac{k^2-1}{k-2}}\left[1 - \frac{\sum_{i=2}^{k}(\bar{Y}_i(b_n) - \bar{Y}_{i-1}(b_n))^2}{\sum_{i=1}^{k}(\bar{Y}_i(b_n) - \bar{X}_n))^2}\right].$$

Sotto l'ipotesi nulla $H_o$ che le medie dei batches siano incorrelate, C≈N(0,1) per *b* elevati, la media dei batch diventa approssimativamente normale, o per *k* elevati, per effetto del teorema del limite centrale [7]. In entrambi i casi, per ciascun istante se il test indica autocorrelazione si usa la regola FNB altrimenti la SQRT. Inoltre, entrambe le procedure danno convergenze analoghe a quelle ottenute con il metodo SQRT.

Nel caso degli OBM, invece, proposti da Meketon e Schmeiser [9], si considera data la dimensione *b* dei batches e si usano *n-b+1* overlapping batches per stimare μ e Var($\bar{X}_n$). Il primo batch consiste nelle osservazioni $Y_1,\ldots,Y_b$ il secondo nelle osservazioni $Y_2,\ldots,Y_{b+1}$. La stima di μ è:

$$\bar{X}_n = \frac{1}{n-b+1}\sum_{i=1}^{n-b+1}\bar{Y}_i(b)$$

con $\bar{Y}_i(b)$ medie dei batches, date da:

$$\bar{Y}_i(b) = \frac{1}{b}\sum_{i=1}^{i+b-1} Y_j \quad \text{con i=1,...,n-b+1}$$

e varianza campionaria

$$S^2 = \frac{1}{n-b}\sum_{i=1}^{n-b+1}(\bar{Y}_i(b) - \bar{X}_n)^2 .$$

Lo stimatore della media risulta essere una media pesata degli stimatori dei batches non sovrapposti. Inoltre, asintoticamente, lo stimatore della varianza così ottenuto e quello della varianza delle medie dei batches, ricavato senza sovrapposizione, hanno lo stesso valore atteso ma il rapporto tra le varianze dei due tende ad essere inferiore a 1. Infine, la varianza dello stimatore $S^2$ è meno sensibile alla scelta della dimensione dei batches rispetto alla varianza della stima della varianza ottenuta per campioni non sovrapposti. Infine, se $b_n$ è definito come nel caso della SQRT e vale l'ASA, allora anche in questo caso il prodotto tra $S^2$ e $b_n$ tende all'infinito alla varianza $\sigma^2_\infty$ .

Date le limitazioni e complessità derivanti dall'impiego delle metodologie del batch means, a partire da queste sono stati sviluppati dei metodi innovati che permettono la valutazione del numero di batches e della loro dimensione in relazione all'ampiezza dell'intervallo di confidenza desiderato. Tra i principali è possibile considerare la procedura di arresto in due fasi sviluppata, in un primo momento, integrando il metodo delle medie dei batches con le teorie di Stein e successivamente ampliata attraverso l'impiego delle serie di tempo standardizzate. La procedura consiste nel determinare la lunghezza della simulazione per sistemi stazionari in modo da ottenere un intervallo di confidenza per lo stimatore della media μ con un'ampiezza predeterminata ε dell'intervallo [126].Questo tipo di valutazione è ottenuta combinando l'approccio di Stein con il metodo batch means. La procedura sviluppata da Stein permette la definizione dell'intervallo di confidenza con una probabilità 1-δ e semiampiezza ε per la media di una variabile normale di varianza $\sigma^2$ incognita. Il metodo consiste nella valutazione della varianza campionaria di un campione $\{X_1,\ldots,X_m\}$ di dimensione $m \geq 2$ estratto dalla popolazione. Il numero di osservazioni necessarie è :

$$N = \max\left\{m, \left\lceil \frac{s^2 t^2_{m-1,\delta}}{\varepsilon^2} \right\rceil\right\}$$

dove $t_{m-1,\delta}$ è il quantile $100(1-\delta/2)\%$ superiore della distribuzione t-Student con *m-1* gradi di libertà, il simbolo $\lceil\ \rceil$ indica il più piccolo intero maggiore o uguale della quantità contenuta all'interno e $S^2 = \frac{1}{m-1}\sum_{i=1}^{m}\left(X_i - \left(\frac{1}{m}\sum_{k=1}^{m} X_k\right)\right)^2$. Noto *N*, si raccolgono ulteriori *N-m* osservazioni e si stima la media campionaria delle *N* osservazioni totali. Si ottiene, così, l'intervallo di confidenza:

$$\left[\frac{1}{N}\sum_{i=1}^{N} X_i - \varepsilon, \frac{1}{N}\sum_{i=1}^{N} X_i + \varepsilon,\right]$$

tale che:

$$P\{\mu \in \left[\frac{1}{N}\sum_{i=1}^{N} X_i - \varepsilon, \frac{1}{N}\sum_{i=1}^{N} X_i + \varepsilon,\right]\} \geq 1-\delta.$$

A partire da tali osservazioni, è possibile applicare il metodo batching [126]. In tal caso, si possono analizzare due tipi di processi: un processo continuo stazionario con output **Y** ={Y(t) : t ≥0}oppure un processo discreto stazionario con output **Y** ={$Y_n$ : n ≥0} entrambi di media μ. I entrambi i casi, si effettua una prima simulazione di ampiezza $1/\varepsilon^2$ e la si divide in $m$≥2 subruns di uguale dimensione $1/m\varepsilon^2$.Si procede, quindi, al calcolo della media delle osservazioni per ogni subrun che per il processo continuo può essere espressa come:

$$\overline{Y}_i(\varepsilon) = \frac{\int_{(i-1)/(m\varepsilon^2)}^{i/(m\varepsilon^2)} Y(s)ds}{1/(m\varepsilon^2)} \quad \text{con i≥1}$$

. Invece nel caso di processo discreto, detta b la dimensione del singolo batch,, per l'$i$-esimosubrun si avrà:

$$\overline{Y}_i(b) = \frac{1}{b}\sum_{j=1}^{b} Y_{j+(i-1)b}$$

Note le $\overline{Y}_i$ , si determina la stima della varianza campionaria come:

$$S^2 = \frac{1}{m-1}\left(\sum_{i=1}^{m}\overline{Y}_i - \frac{1}{m}\sum_{j=1}^{m}\overline{Y}_j\right)^2$$

A questo punto, il numero ottimale di subrun necessari è $N_a(\varepsilon) = \max\left\{m, \left\lceil \frac{s^2 t^2_{m-1,\delta}}{\varepsilon^2} \right\rceil\right\}$. Si devono ottenere, quindi, ulteriori $N_a$-$m$ batches di dimensione $1/m\varepsilon^2$, ovvero la lunghezza ottimale della simulazione è $N_a/m\varepsilon^2$.L'intervallo di confidenza ottenuto è:

$$I_a = \left[\frac{1}{N_a}\sum_{i=1}^{N_a}\overline{Y}_i - \varepsilon, \frac{1}{N_a}\sum_{i=1}^{N_a}\overline{Y}_i + \varepsilon,\right]$$

Tale intervallo risulta asintoticamente valido, ovvero:

$$\lim_{\varepsilon\to 0} P\{\mu \in \left[\frac{1}{N_a}\sum_{i=1}^{N_a}\overline{Y}_i - \varepsilon, \frac{1}{N_a}\sum_{i=1}^{N_a}\overline{Y}_i + \varepsilon,\right]\} \geq 1-\delta.$$

L'asintotica validità permette di non assumere quale ipotesi di base la normalità della distribuzione di partenza, ma solo che valga il teorema del limite centrale. Nel caso in cui, invece, si voglia ottenere una precisione relativa dell'intervallo di

confidenza, come il 10% del valore stimato, la procedura è inalterata ma si ha che:

$$N_r = \max\left\{ m, \left\lceil \frac{S^2 t^2_{m-1,\delta}}{\varepsilon^2 \left( \frac{1}{m} \sum_{i=1}^{m} \overline{Y}_i \right)^2} \right\rceil \right\}$$

$$I_r = \left[ \frac{1-\varepsilon}{N_r} \sum_{i=1}^{N_r} \overline{Y}_i(b), \frac{1+\varepsilon}{N_r} \sum_{i=1}^{N_r} \overline{Y}_i(b) \right]$$ nel caso discreto;

$$I_r = \left[ \hat{\mu}_r(\varepsilon) - \varepsilon \left| \hat{\mu}_r(\varepsilon) \right|, \hat{\mu}_r(\varepsilon) + \varepsilon \left| \hat{\mu}_r(\varepsilon) \right| \right]$$ per i processi continui

con $\hat{\mu}_r(\varepsilon) = \frac{1}{N_r(\varepsilon)} \sum_{i=1}^{N_r(\varepsilon)} \overline{Y}_i$ .

Naturalmente è possibile implementare la stessa procedura anche nel caso di processi continui.

Il procedimento può essere modificato valutando, non la stima della varianza ma quella della varianza asintotica. Tale analisi può essere effettuata impiegando il metodo delle serie di tempo standardizzate. Per applicare tale tecnica è necessario ipotizzare la validità del teorema del limite centrale funzionale. Si consideri $\{Z_\varepsilon : \varepsilon>0\}$ una famiglia di elementi casuali aventi valori nello spazio delle funzioni continue $C[0,\infty)$ .Se Z è un elemento casuale di C, $Z_\varepsilon$ converge debolmente a Z Se $Ef(Z_\varepsilon) \rightarrow Ef(Z)$ quando $\varepsilon \rightarrow 0$ per ogni funzione continua limitata $f:C[0, \infty) \rightarrow R$. Se **Y**={Y(t): t≥0} Є D[0,∞), insieme delle funzioni a valore reale continue a destra e con limite a sinistra, è il processo stocastico rappresentante l'output della simulazione a valori reali, si definisce $Z_\varepsilon(t) = \frac{1}{\varepsilon}\left(\overline{Y}_\varepsilon(t) - \mu t\right)$ con $\overline{Y}_\varepsilon(t) = \frac{\int_0^{t/\varepsilon^2} Y(s)ds}{\frac{1}{\varepsilon^2}}$

per t≥0. nel caso discreto, è possibile utilizzare la stessa procedura ponendo

Y(t)=$Y_{\lfloor t \rfloor}$. Il teorema è il seguente: Esiste un valore costante finito μ e σ tale che $Z_\varepsilon$ $\rightarrow \sigma B$ quando $\varepsilon \rightarrow 0$ con B, moto Browniano standard. Da questo teorema si evince che:

$$\frac{\int_0^t Y(s)ds}{t} - \mu = \frac{1}{\sqrt{t}} Z_{\frac{1}{\sqrt{t}}}(1) \Rightarrow 0 * \sigma B(1) = 0 .$$

Ovvero, che μ è la media da stimare stazionaria. Inoltre, si ha che:

$$\sqrt{t}\left[\frac{\int_0^t Y(s)ds}{t} - \mu\right] = Z_{1/\sqrt{t}}(1) \Rightarrow \sigma B \qquad \text{con } t \rightarrow \infty.$$

Poiché B(1) è una normale di media nulla e varianza unitaria, $\sigma^2$ è la varianza asintotica di **Y**. La procedura è differente rispetto alla precedente perché non si valuta la stimma della varianza per i primi *m* subruns ma si definisce la funzione *g* dipendente da m con le seguenti caratteristiche:

1. per ogni $z \in C[0,\infty)$ la g(z) dipende solo da $\{z(s): 0 \leq s \geq 1\}$;
2. $g(\alpha z) = \alpha g(z)$ per $\alpha > 0$ e $z \in C[0,\infty)$;
3. $g(z-\beta k) = g(z)$ per $\beta \in R$ e $z \in C[0,\infty)$, dove k(t)=t;
4. $P\{g(B)>0\}=1$;
5. $P\{B \in G(g)\}=0$;
6. g(B) ha una funzione di distribuzione continua.

La prima condizione assicura che g dipende solo dall'evoluzione del processo al tempo 1; applicando *g* al processo $Z_\varepsilon$, $g(Z_\varepsilon)$ è unicamente determinata da $\{Z_\varepsilon(s) : 0 \leq s \geq 1\}$, che corrisponde all'intervallo di tempo tra 0 e $1/\varepsilon^2$ del processo originario **Y.** Quindi, *g* dipende solo dall'evoluzione del processo nel primo stadio. La seconda condizione assicura che una modifica dell'unità di misura dell'output della simulazione anche lo stimatore basato sulla funzione *g* riflette tale cambiamento. Infine, la terza condizione implica che $g(Z_\varepsilon)$ non dipende dal parametro ignoto μ e le ultime due sono condizione tecniche. La procedura consiste nell'effettuare una prima simulazione di lunghezza $1/\varepsilon^2$ divisa in $m \geq 1$ subruns e nel calcolo di $s(\varepsilon)=m^{1/2}\varepsilon g(Z_\varepsilon)$, ovvero della stima della deviazione

standard σ, che dipende dall'evoluzione del processo solo nel primo stato. Si definisce, quindi, il numero ottimale di batches come:

$$N_a = \max\left\{m, \left\lceil \frac{s^2(\varepsilon)a_\delta^2}{\varepsilon^2} \right\rceil\right\}$$

Con $a_\delta$ pari al 100(1-δ/2)% quantile della variabile casuale B(1)/g(B), che esiste in virtù del fatto che tale funzione è continua e strettamente crescente. Si effettua, quindi, la simulazione dei restanti $N_a$-$m$ batches di dimensione $1/m\varepsilon^2$, ottenendo un intervallo di confidenza pari a :

$$I_a = \left[ \frac{1}{N_a(\varepsilon)} \sum_{i=1}^{N_a(\varepsilon)} \overline{Y}_i(\varepsilon) - \varepsilon, \frac{1}{N_a(\varepsilon)} \sum_{i=1}^{N_a(\varepsilon)} \overline{Y}_i(\varepsilon) + \varepsilon, \right]$$

Se vale il teorema funzionale del limite centrale si dimostra che le seguenti asserzioni sono vere:

1. $N_a(\varepsilon) \to N_a$ se $\varepsilon \to o$ con $N_a$ un'adeguata variabile casuale di limite;
2. $\lim_{\varepsilon \to 0} P\{\mu \in I_a(\varepsilon)\} \geq 1 - \delta$.

È sperimentalmente provato, tuttavia, che nel caso in cui sia necessario effettuare il secondo step della procedura la lunghezza della simulazione risulta molto elevata. Al fine di ridurre tale lunghezza è possibile modificare ulteriormente la procedura ottenendo un procedimento che assicura una lunghezza mai superiore e spesso inferiore a quella ottenuta dal metodo precedente. Tale procedimento consiste nel modificare il secondo stadio procedendo non con la simulazione dei restanti $N_a$-$m$ batches, ma con la definizione di $Q_a$ data da:

$$Q_a = \max\left\{m, \frac{s^2(\varepsilon)a_\delta^2}{\varepsilon^2}\right\},$$

Tale parametro differisce da $N_a$ perché non presenta il limite superiore. Si effettua, quindi, la simulazione da $1/\ \varepsilon^2$ a $Q_a(\varepsilon)/(m\ \varepsilon^2)$ e si considera l'intero intervallo come un unico batch finale. A questo punto, note le $\overline{Y}_i(\varepsilon)$ per i=1,..,m si considera media campionaria del processo nel secondo stadio:

$$\overline{Y}'_{m+1}(\varepsilon) = \frac{\int_{m/(m\varepsilon^2)}^{Q_a(\varepsilon)/m\varepsilon^2} Y(s)ds}{(Q_a(\varepsilon)-m)/(m\varepsilon^2)}$$

noto che il batch corrispondente a tale Y può non presentare la stessa dimensione degli altri *m* batches. Si definisce, quindi, quale intervallo di confidenza:

$$I'_a(\varepsilon) = [\hat{\mu}_a(\varepsilon)-\varepsilon, \hat{\mu}_a(\varepsilon)+\varepsilon],$$

con $$\hat{\mu}_a = \frac{1}{Q_a(\varepsilon)}\left[\sum_{i=1}^{m}\overline{Y}_i(\varepsilon) + (Q_a(\varepsilon)-m)\overline{Y}'_{m+1}(\varepsilon)\right].$$

Il vantaggio di tale metodo è che se $N_a(\varepsilon)>m$ si ottiene :

$$N_a(\varepsilon)=\lceil Q_a(\varepsilon)\rceil,$$

il che comporta che l'utilizzo di $Q_a$ non richiede di effettuare la simulazione da $Q_a(\varepsilon)/\varepsilon^2$ a $\lceil Q_a(\varepsilon)\rceil / \varepsilon^2$. Anche in questo caso, sono verificate le asserzioni 1 e 2 definite precedentemente. Come nel caso del metodo del batch means, si può considerare un intervallo relativo, sia utilizzando $N_r$ che $Q_r$, definiti rispettivamente come segue:

$$N_r(\varepsilon) = \max\left\{m, \left\lceil \frac{S^2(\varepsilon)a_\delta^2}{\varepsilon^2\left(\frac{1}{m}\sum_{i=1}^{m}\overline{Y}_i(\varepsilon)\right)^2}\right\rceil\right\},$$

con $$\hat{\mu}_r(\varepsilon) = \frac{1}{N_r(\varepsilon)}\sum_{i01}^{N_r(\varepsilon)}\overline{Y}_i(\varepsilon), \quad ;$$

$$I_r(\varepsilon) = \left[\hat{\mu}(\varepsilon)-\varepsilon\left|\hat{\mu}(\varepsilon)\right|, \hat{\mu}(\varepsilon)+\varepsilon\left|\hat{\mu}(\varepsilon)\right|\right];$$

Per $Q_r$ si ha, invece:

$$Q_r(\varepsilon) = \max\left\{ m, \frac{S^2(\varepsilon) a_\delta{}^2}{\varepsilon^2 \left( \frac{1}{m} \sum_{i=1}^{m} \overline{Y}_i(\varepsilon)^2 \right)} \right\}$$

e $$I_r^{'}(\varepsilon) = \left[ \hat{\mu}_r^{'}(\varepsilon) - \varepsilon \left| \hat{\mu}_r^{'}(\varepsilon) \right|, \hat{\mu}_r^{'}(\varepsilon) + \varepsilon \left| \hat{\mu}_r^{'}(\varepsilon) \right| \right];$$

con $$\hat{\mu}_r^{'}(\varepsilon) = \frac{1}{Q_r(\varepsilon)} \left[ \sum_{i=1}^{m} \overline{Y}_i(\varepsilon) + (Q_r(\varepsilon) - m) \overline{Y}_{m+1}^{r}(\varepsilon) \right].$$

In entrambi i casi valgono le asserzioni 1 e 2. Naturalmente, l'utilizzo di tale procedura richiede la scelta di una funzione *g* adeguata, scelta che spesso risulta molto complessa.

Sia nel caso in cui ci si riferisca al metodo del batch means, che nel caso in cui si impieghino le serie di tempo standardizzzate, il problema fondamentale è quello di definire il numero m di batches iniziali. Nel caso in cui si abbia un'idea del valore di $\sigma^2$ prima di effettuare il campionamento, è possibile fare riferimento alle tabelle fornite da Seelbinder. Inoltre, Schmeiser suggerisce che il numero totale di batches da usare nel metodo del batch means non sequenziale per una fissata lunghezza di simulazione debba essere piuttosto ridotto, in genere da 10 a 30. In questo modo, l'implicita assunzione che i batches siano i.i.d. e normali è maggiormente soddisfatta. Usando una procedura in due stadi, quindi è possibile considerare un numero ancora più ristretto di batches iniziali, ad esempio da 5 a 15.

Tutte le procedure esposte considerano la valutazione di un solo parametro, tuttavia può risultare, in alcuni casi, necessario considerare le stime dei valori medi di numerosi parametri, spesso correlati. Se $D_i$ è il livello di confidenza 1-α per i parametri $\mu_i$ con i=1,2,..,k; allora si ha che:

$P(\cap^{k}_{i=1}\{ \mu_i \in D_i\}) \geq 1-\sum^{k}_{i=1}\alpha_i$.

Se il livello di confidenza totale deve essere al massimo 1-α, le $\alpha_i$ possono essere scelte in modo che la loro somma sia pari ad α. Oltre alla possibilità di combinare come illustrato i livelli di confidenza sulla base della tecnica del batch means, sono state sviluppate ulteriori procedure innovative nel caso di analisi di più parametri [9].

Uno dei metodi ampiamente diffusi è quello sviluppato da Becker et al., valido solo per sistemi del secondo ordine asintoticamente stabili [23]. Si considerino alcuni criteri di performance $C_i$ e per ognuno un livello di confidenza $1-\eta_i$. L'algoritmo calcola gli intervalli di confidenza a un certo istante $T$ e si arresta quando tutti gli intervalli trovati sono minori di quelli specificati. In particolare, si effettua una prima simulazione di lunghezza $T_1$ e si calcola l'intervallo di confidenza. L'ipotesi è che il processo sia strettamente ergodico. Anche in questo caso è possibile considerare due tipi di processo; uno Y(t) funzione del tempo, l'altro $Y_k$ funzione di k, numero di eventi. La media temporale del processo Y(t) , nella simulazione a eventi discreti con $T=t_p$ è

$$\overline{Y}_T = \frac{1}{t_p}\sum_{i=1}^{p} Y(t_i)(t_{i+1}-t_i)$$

dove $p$ è il numero di eventi che modificano la funzione ad ogni $t_i$. La stretta ergodicità implica che $\mathrm{E}\left[\frac{1}{t_p}\sum_{i=1}^{p} V(Y(t_i))(t_{i+1}-t_i)\right] \underset{t_p\to\infty}{\to} \mu$

per ogni funzione V di Y(t) con $\mu = \lim_{t\to\infty} E[V(Y(t_i))]$.

Un esempio di variabile simile è il numero di pezzi presenti in un buffer o in coda ad una macchina. Si considerano anche variabili definite non in funzione del tempo ma solo quando alcuni eventi si verificano. In questo caso,

$$\overline{Y}_N = \frac{1}{N}\sum_{k=1}^{N} Y_k$$

con $N$ numero di diversi valori di $Y_k$. In questo caso, la stretta ergodicità implica che per ogni funzione V di $Y_k$

$$E\left[\frac{1}{N}\sum_{k=!}^{N} V(Y_k)\right] \underset{T\to\infty}{\to} \mu \qquad \text{con} \qquad \mu = \lim_{t\to\infty} E[V(Y_k)].$$

Un esempio di tale tipo di processo è il trasferimento di un pezzo da un centro di produzione ad un altro. Poiché $T$, lunghezza della simulazione, e $N$ tendono ad essere proporzionali, ovvero $N/T$ tende alla densità di eventi, si considererà solo la valutazione della lunghezza $T$ ottimale. L'intervallo di confidenza di ampiezza ε con livello di confidenza pari a 1-η è tale che $P(|\overline{Y}_T - \mu| < \varepsilon) \geq 1-\eta$, dove $\overline{Y}_T$ è la

media campionaria e μ è il valore atteso E{T(t)} che si vuole stimare. Applicando alla probabilità il teorema di Chebyshev si può calcolare ε dato che:

$$P\{|\overline{Y}_T - \mu| < \varepsilon\} > 1 - \sigma^2(\overline{Y}_T)/\varepsilon^2 \text{ da cui } \varepsilon = \sigma(\overline{Y}_T)/\sqrt{\eta}\,.$$

dove $\sigma^2(\overline{Y}_T)$ è la varianza dello stimatore e non la varianza del processo. Quindi, noto η è necessario determinare la lunghezza della simulazione *T* per la quale si ottenga una deviazione standard tale che valga questa relazione. Nell'ipotesi di processo del secondo ordine asintoticamente stabile e strettamente ergodico, si ha che $\sigma^2(\overline{Y}_T) \approx A/T$ con A>0 costante.Tuttavia, né $\sigma^2(\overline{Y}_T)$ né A sono direttamente calcolabili. Per questo motivo, la varianza è ottenuta dividendo in *n* batches la simulazione, dopo aver preventivamente eliminato il transitorio, con durata dei subrun pari a $T_b = T/n$. Poiché $\sigma^2(\overline{Y}_{T_b}) \approx A/T_b$, si ha che $\sigma^2(\overline{Y}_T) = \sigma^2(\overline{Y}_{T_b})/n$ dove $\sigma^2(\overline{Y}_{T_b})$ è stimata come:

$$\frac{1}{n-1}\sum_{i=1}^{n} \overline{Y}_{T_b}^{\,2} - n\overline{Y}_T^2$$

con $\overline{Y}_{T_b} = \frac{1}{T_b}\int_{(i-1)T_b}^{iT_b} X(t)dt$ media dell'*i*-esimo batch e $\overline{Y}_T = \frac{1}{n}\sum_{i=1}^{n}\overline{Y}_{T_b}$, da cui:

$$\varepsilon = \frac{1}{\sqrt{\eta}}\frac{\sigma(\overline{Y}_{T_b})}{\sqrt{\eta}}.$$

Noti i parametri $C_i$ e fissato il livello di confidenza $1-\eta_i$ si calcola il valore $\varepsilon_{i,T}$ all'istante *T* o il valore relativo $\rho_{i,T} = \varepsilon_{i,T}/Z_{i,T}$ e li si confronta con $\varepsilon_i$ o con $\rho_i$. Il primo step dell'algoritmo consiste nel dimensionare $T_b$ per la prima iterazione. Tale valore è ottenuto in modo da avere stimatori significativi e, allo stesso tempo, una simulazione non eccessivamente lunga. A tale fine si utilizzano dei contatori con predefiniti valori limite per ogni parametro. Quando il valore limite è raggiunto, si interrompe la simulazione e l'istante di arresto è scelto come primo $T_b$. Essendo $T = nT_b$, si considera la varianza di ciascun stimatore $\sigma^2_{i,T} \approx A_i/T$ e $\varepsilon_{i,T} \approx B_i/\sqrt{T}$ con $B_i$ costante. Se $\varepsilon_{i,T} < \varepsilon_i$ per ogni criterio l'algoritmo si arresta, altrimenti si determina il δi, ovvero il tempo necessario per ottenere $\varepsilon_i$. Da cui, in corrispondenza di δi si ha che $\varepsilon_{i,\,\delta} = \varepsilon_i = B_i/\sqrt{\delta i}$ e, quindi, posto $B_i = \varepsilon_{i,\,T} * \sqrt{T}$ si ha che:

$$\delta i = \left( \frac{\varepsilon_{i,T}}{\varepsilon_i} \right)^2 T .$$

Per ottenere l'adeguato intervallo di confidenza per ogni parametro si sceglierà, quindi, come lunghezza della simulazione

$T' = \mathrm{Max}_i(\delta i)$ oppure

$$T' = \left[ Max_i \left( \frac{\varepsilon_{i,T}}{\varepsilon_i} \right) \right]^2 T .$$

A questo punto, noto T' si considera la dimensione ottimale dei batch come $T'_b = \theta T_b$ con $\theta = T/T'$. Si può, quindi, rifinire ulteriormente l'algoritmo con tecniche , quali quella del reimpiego dei batch già ottenuti. In altri termni, invece di simulare ulteriori *n* batches di dimensione $T'_b$ da *T* a *T*+*T'* è possibile considerare per $\theta$ ridotti un'unica simulazione di lunghezza *T'*. Poichè, in altri termini, i valori di interesse sono delle medie sui batches, se $\theta$ è intero, è possibile ottenere i valori necessari dalle medie dei batches già simulati. Poiché non è detto che $\theta$ sia intero si considera $\beta = \lceil \theta \rceil$ e $T' = \beta T$. È necessario distinguere tre casi:

1) ***n è un multiplo di β;***
2) ***n non è multiplo di β;***
3) ***β è maggiore di n.***

Nel primo caso, è possibile effettuare solo la simulazione di (n-n/ β) batches nuovi fino a *T'* e calcolare le medie per gli $n/\beta$ subrun non simulati come:

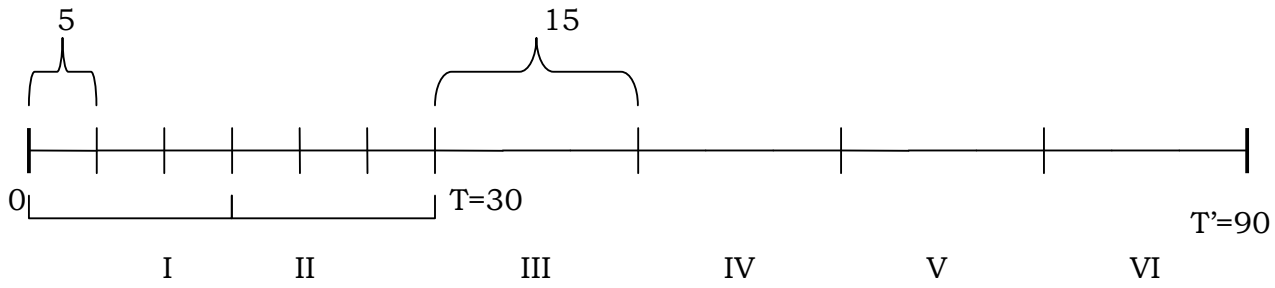
$$A'_1 = 1/\beta \, (A_1 + A_2 + \ldots + A_\beta)$$

$$\ldots\ldots\ldots\ldots$$

$$A'_{n/\beta} = 1/\beta \, (A_{(n/\beta - 1)\beta} + \ldots\ldots + A_{(n/\beta)\beta}).$$

Ad esempio, nel caso in cui T sia pari a 30 min, n a 6 e β a 3, si ha che $T_b$ è pari a 5 min, T' a 90 min e $T'_b$ è uguale a 15 min. Invece di effettuare una seconda simulazione di lunghezza 90 min e ripartirla in 6 batches è possibile effettuare la simulazione da T a T', ovvero di durata 60 min, ripartendo questa in n-(n/ β )=4

subruns di lunghezza 15 min. I valori medi degli altri due batches sono ottenuti a partire dai 6 batches precedentemente ottenuti secondo la $A'_i$.




Nel secondo caso, invece, se $n$ non è un multiplo di $\beta$, non sono effettuati $\left\lfloor \frac{n}{\beta} \right\rfloor + 1$ nuovi batches. In breve, si effettuerà una simulazione da $T$ a $T'$ con un primo batch di dimensione $T_b$ che sarà impiegato con quelli precedenti per calcolare i valori medi dei $\left\lfloor \frac{n}{\beta} \right\rfloor + 1$ non simulati e poi si simulano i restanti n-$\left\lfloor \frac{n}{\beta} \right\rfloor + 1$ di dimensione $T'_b$.

Infine, se $\beta$ è maggiore di $n$, significa che $T$ è troppo ridotto, addirittura minore del nuovo batch, e nessun batch può essere reimpiegato. In questo caso, la simulazione fino a $T$ non è considerata e si effettua la simulazione da $T$ a $T+T'$ e la si suddivide in $n$ nuovi batches. Tale situazione dipende da contatori con valori limite troppo ridotti. Inoltre, se il primo $T_b$ è troppo ridotto si ottiene un valore di $\beta$ molto elevato e errato. Per questo motivo, in alcuni casi si suggerisce l'impiego di un valore limite superiore per $\beta$, al fine di evitare simulazioni inutilmente lunghe. Tuttavia l'impiego di contatori evita tale problema. Il reimpiego dei batches non può essere effettuato nel caso di più iterazioni. Quindi, al vantaggio derivante dalla eliminazione di un unico transitorio nel caso della tecnica batch si affianca anche la possibilità di riutilizzare le informazioni ottenute nella prima

simulazione. Se, invece, si ha che $\beta=2$, ovvero assume il valore minimo per il quale si prosegue nella definizione di *T'*, non si modifica la dimensione dei batches ma il numero. In altri termini, detto *n* il numero di subrun di default, si simula il *(n+1)*-esimo batch e si confronta la $(\varepsilon_{i,(n+1)Tb})_i$ con $(\varepsilon_i)_i$. Se necessario, si prosegue con la simulazione del *(n+2)*-esimo batch e si ripete il confronto. Per evitare che tale metodo richieda un numero troppo elevato di iterazioni, si ripete al massimo per *n* nuovi batches, ottenendo, così, una lunghezza di simulazione di *2T*. Se il valore limite dell'intervallo di confidenza non è ancora raggiunto, si procede con la costituzione tradizionale dei batches.

Infine, è necessario considerare la possibilità che l'algoritmo non converga. In genere, tale evenienza si verifica per intervalli di confidenza troppo ridotti, livelli eccessivamente elevati oppure nel caso in cui non siano verificate le ipotesi, ad esempio se il sistema risulta instabile. Per questo motivo, di norma, si definisce come limite massimo di iterazioni realizzabili pari a quattro.

Fino a questo punto, si è focalizzata l'attenzione sulla scelta della lunghezza della simulazione ottimale per un singolo sistema. In molti casi, tuttavia, la simulazione è impiegata per confrontare diverse possibili configurazioni di uno stresso sistema al fine di selezionare l'assetto ottimale. Tale problema risulta in qualche modo connesso con la determinazione della lunghezza della simulazione, in quanto anche in questo caso, la scelta della configurazione ottimale sarà effettuata con un certo livello di confidenza connesso alla durata della simulazione stessa per i diversi designs. Come nel caso della simulazione di un unico sistema, è possibile applicare il metodo del batch means. Tale procedura può essere impiegata laddove si desideri scegliere, tra *k* diversi sistemi, quello che presenta la maggiore o minore media stazionaria di un dato parametro con varianze asintotiche anche differenti per i diversi assetti [SIM5]. Nell'ipotesi che il sistema migliore sia quello con media più alta, l'obiettivo è la scelta della configurazione migliore con una data probabilità attraverso l'individuazione di una zona di indifferenza, ovvero di un intervallo di ampiezza $\delta$ con estremo superiore dato dalla maggiore media stimata, all'interno del quale qualunque sistema può essere scelto indifferentemente. Inoltre, il metodo permette la definizione degli intervalli di confidenza, denominati *multiple comparisons with the best* (MCB, per $\mu_i$-$max_{j\neq i}\mu_j$ con i=1,2,..,k.

Si consideri $\mathbf{Y_i}$={$Y_i$(t): t≥0}o, nel caso di sistemi discreti $\mathbf{Y_i}$={$Y_{i,l}$:l=0,1,2,…} ricavabile ponendo $Y_i(t)= Y_{i,\lfloor t \rfloor}$,il processo stocastico rappresentante l'output del sistema i-esimo e si assuma che i processi siano indipendenti tra loro. Si definisce il vettore $\mathbf{Y}=(\mathbf{Y_1},\mathbf{Y_2},\ldots,\mathbf{Y_k})$ e $Y(t)=(Y_1(t),Y_2(t),\ldots,Y_k(t))$ e si assume che il processo stocastico soddisfi il teorema del limite centrale funzionale. Tale assunzione comporta l'ipotesi che esista una matrice ∑ non singolare kXk, che in questo caso è la matrice di covadianza ed è una matrice diagonale, e una costante $\mu=(\mu_1,\mu_2\ldots..,\mu_k)$ Є $R^k$ tale che

$$X_\delta \rightarrow \sum B \text{ quando } \delta \rightarrow 0$$

dove B è un moto Browniano standard k-dimensionale e $X_\delta=(X_{1,\delta}, X_{2,\delta},\ldots., X_{k,\delta})$ con

$$X_{i,\delta} = \frac{1}{\delta}\left( \frac{\int_0^{t/\delta^2} Y_i(s)ds}{1/\delta^2} - \mu_{i,t} \right), \qquad \text{per } t\geq 0.$$

In genere, gran parte dei sistemi reali soddisfano tale teorema. Ordinate le medie in ordine crescente $\mu_{(1)}\leq \mu_{(2)}\leq\ldots.. \leq\mu_{(k)}$, la zona di indifferenza è l'intervallo ($\mu_{(k)}$-δ, $\mu_{(k)}$]. La procedura è realizzata in due fasi. Nella prima fase, si effettuano indipendentemente le simulazioni di ciascun sistema di durata $T_i=T_i(\delta)$ proporzionale a $1/\delta^2$. Per ogni sistema, gli output sono raggruppati in *m* batches di dimensione $T_i/m$ e si considera la media campionaria per ciascun batch *j*-esimo come:

$$Z_{i,j} = \frac{1}{T_i/m} \int_{(j-1)T_i/m}^{jT_i/m} Y_i(s)ds \text{ con } j\geq 1.$$

La stima della varianza della media campionaria totale può essere ottenuta come:

$$S_i^2 = \frac{1}{m-1}\left( \sum_{j=1}^{m} Z_{i,j} - \frac{1}{m}\sum_{k=1}^{m} Z_{i,k} \right)^2,$$

in funzione della quale si determina per ogni sistema il numero totale di batches ottimale, attraverso la seguente relazione:

$$N_{\alpha,i(\delta)} = \max\left\{m, \left\lceil \frac{S_i^2 a_\delta^2}{\delta^2} \right\rceil\right\}.$$

Si procede, quindi, alla realizzazione della simulazione per ogni sistema *i* dei restanti $N_{a,i(\delta)} - m$ batches di dimensione $T_i/m$ e si calcolano le stime delle medie per questi ultimi. Il calcolo della media totale è, quindi:

$$\hat{\mu}_{\alpha,i} = \frac{1}{N_{\alpha,i(\delta)}} \sum_{j=1}^{N_{\alpha,i(\delta)}} Z_{i,j}$$

Allo stesso tempo, si seleziona la media stimata più alta e si costruisce l'intervallo MCB di confidenza dato da:

$$I_{\alpha,i(\delta)} = \left[ \left( \hat{\mu}_{\alpha,i} - \max_{j \neq i} \hat{\mu}_{\alpha,j} - \delta \right)^{-}, \left( \hat{\mu}_{\alpha,i} - \max_{j \neq i} \hat{\mu}_{\alpha,j} + \delta \right)^{+} \right]$$

dove $(x)^-$ indica il min (0,x) e $(x)^+$ indica il max (x,0). Se si definisce un parametro $k_\alpha$, con range di variazione $1 \leq k_\alpha(\delta) \leq k$, tale che :

$$\mu_{(k)} - \mu_{(l)} < \delta \quad \text{per tutte le } l \geq k_\alpha(\delta)$$

$$\mu_{(k)} - \mu_{(i)} \geq \delta \quad \text{per tutte le } i < k_\alpha(\delta)$$

è possibile affermare che selezionando un qualunque sistema $(k_\alpha(\delta))$, $(k_\alpha(\delta)+1),\ldots,(k)$, l'obiettivo è soddisfatto allo stesso modo, ovvero tali valori sono i soli compresi nell'intervallo di indifferenza. Per valutare la validità asintotica del metodo, si definiscano gli eventi:

$$CS_\alpha(\delta) = \left\{ \max_{l \geq k_\alpha(\delta)} \hat{\mu}_{\alpha,(l)}(\delta) > \max_{i < k_\alpha(\delta)} \hat{\mu}_{\alpha,(i)}(\delta) \right\},$$

che è l'evento di corretta selezione e

$$JC_\alpha(\delta) = \left\{ \mu_i - \max_{j \neq i} \mu_j \in I_{\alpha,i}(\delta), i = 1,2,\ldots,k \right\},$$

che indica che tutte le differenze effettive tra le medie dei singoli sistemi e il valore massimo sono contemporaneamente coperte dai loro intervalli di confidenza MCB. Si verifica, nell'ipotesi in cui valga il teorema funzionale del limite centrale, che:

$$\lim_{\delta \to 0} P\{CS_\alpha(\delta) \cap JC_\alpha(\delta)\} \geq 1 - \alpha$$

Quindi, tale metodo risulta asintoticamente valido.

Nel caso in cui si debba considerare un'ampiezza della zona di indifferenza relativa, tale intervallo può essere definito come $(\mu_{(k)}-\delta|\ \mu_{(k)}\ |,\ \mu_{(k)}]$. In questo caso, la procedura è la stessa, tuttavia, il numero ottimale di batches è:

$$N_{r,i(\delta)}= \max\left\{m, \left\lceil \frac{S_i^2 a_\delta^2}{\delta^2 \hat{\mu}} \right\rceil\right\},$$

dove $\hat{\mu}$ è il massimo tra le medie totali stimate per i diversi sistemi. In altri termini, per ogni disegno, si considera $\hat{\mu}_{r,i} = \frac{1}{N_{r,i(\delta)}} \sum_{j=1}^{N_{r,i(\delta)}} Z_{i,j}$ e si seleziona il sistema con il valore medio stimato più elevato. Allo stesso modo, l'intervallo di confidenza MCB ottenuto è:

$$I_{r,i(\delta)} = \left[ \left( \hat{\mu}_{r,i} - \max_{j\neq i} \hat{\mu}_{r,j} - \delta \max_{j\neq i} \left|\hat{\mu}_{r,j}\right| \right)^{-}, \left( \hat{\mu}_{r,i} - \max_{j\neq i} \hat{\mu}_{r,j} + \delta \left|\hat{\mu}_{r,j}\right| \right)^{+} \right].$$

Come nel caso dell'intervallo assoluto, è possibile definire il parametro $1\leq k_r(\delta)\leq k$, al fine di individuare l'intervallo di indifferenza, tale che :

$$\mu_{(k)}- \mu_{(l)} <\delta\ |\ \mu_{(k)}|\quad \text{per tutte le } l\geq k_r(\delta)$$

$$\mu_{(k)}- \mu_{(i)}\geq\delta|\ \mu_{(k)}\ |\qquad \text{per tutte le } i<k_r(\delta)$$

Anche in questo caso, sotto l'ipotesi di validità del teorema funzionale del limite centrale,è possibile definire i due eventi questo punto gli eventi $CS_r(\delta)$ e $JC_r(\delta)$ e si verifica che:

$$\lim_{\delta\to 0} P\{CS_r(\delta)\cap JC_r(\delta)\}\geq 1-\alpha.$$

La procedura sviluppata richiede, sia nel caso di intervalli assoluti che relativi, la necessità di definire i parametri $\delta$ e $m$. Poiché tale procedura può essere intesa come un'estensione a più sistemi del metodo in due fasi di stima dei parametri, è possibile riferirsi a tale tecnica nella definizione di tali grandezze. In generale, si assume $5\leq m\leq 15$ e $\delta<0.025$.

Un metodo alternativo consiste nel determinare quali configurazioni possono essere considerate migliori prima di effettuare tutte le $T$ replicazioni, al fine di impiegare diverse replicazioni per ciascuna alternativa. Nel caso in cui sia necessario effettuare il confronto tra più alternative, infatti, se $n$ è il numero totale di diverse configurazioni e $T$ il numero totale di replicazioni o la lunghezza

totale della simulazione o il numero di batches, a seconda dei metodi impiegati, per ciascuna di esse si verifica che l'onere computazionale risulta molto elevato. Un confronto tra i metodi tradizionali e quello proposto è illustrato in figura [SIM7]:

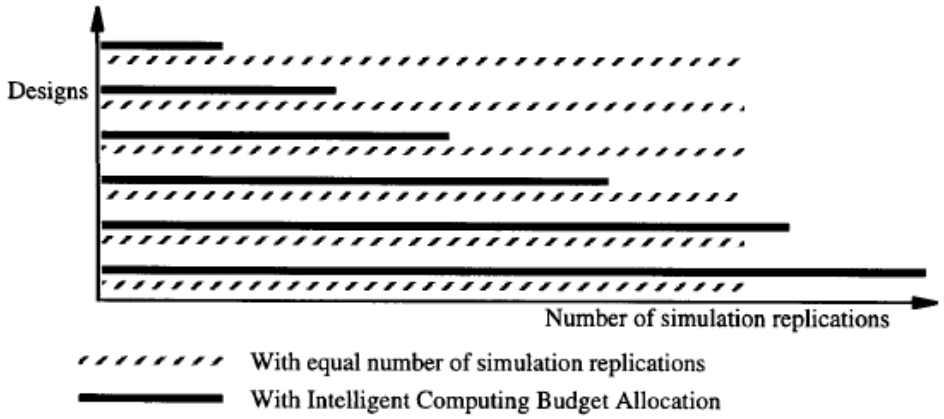



**Figura 4.6 Confronto tra metodi tradizionali ed il metodo proposto**

Si definisca la probabilità dell'evento P{CS}=P{selezione corretta}, ovvero la probabilità che il design con la media campionaria più alta $\hat{\mu}_b$, nell'ipotesi di massimizzazione, sia effettivamente il design migliore, attraverso la sua stima $\prod_{\substack{i=1 \\ i\neq b}}^{k} P\{\hat{\mu}_b > \hat{\mu}_i\}$=APCS. [39].Tale stima può essere effettuata nell'ipotesi di indipendenza dei risultati delle simulazioni dei diversi sistemi e di normalità delle variabili $\hat{\mu}_i \sim$ N( $1/T_i \sum^{N}_{j=1} X_{i,j}$, $\sigma^2_i/T_i$). L'obiettivo è quello di determinare le singole $T_i$ tali da soddisfare il seguente problema di minimizzazione:

$$\min (T_1+\ldots\ldots+T_n),$$

$$\text{s.t. APCS} \geq P^*$$

dove P* è un valore limite inferiore. Le principali difficoltà nella risoluzione di tale problema sono che APCS può essere definita solo dopo aver effettuato le simulazioni e che, essendo i $T_i$ interi, per $n$ elevato lo spazio combinatoriale è molto ampio. Per risolvere tali problematiche è stato sviluppato un algoritmo sequenziale che porta ad una soluzione che in alcuni casi risulta sub-ottima [SIM7]. L'algoritmo consiste nella realizzazione della simulazione di ciascun sistema per una durata $t_0$, sulla base della quale si determina per quale assetto

effettuare un'ulteriore simulazione di ampiezza $\Delta$. In altri termini, si stima la distribuzioni a posteriori all'istante $t_0+\Delta$ sulla base della distribuzione a $t_0$ come

$$N( 1/t_0 \sum^{t0}_{j=1} X_{i,j}, \sigma^2_i/t_0+\Delta).$$

Tale approssimazione è tanto più valida quanto più $t_0$ è ampio e $\Delta$ è ridotto. In questo modo, è possibile stimare APCS a $t_0+\Delta$ , tale stima è definita come EAPCS. Ad ogni step k=1,2…. il problema risulta, quindi, essere :

$$MAX_{\Delta i}\ EAPCS(T^k_1+\Delta_1^k,\ldots..,\ T^k_n+\Delta_n^k)$$

$$s.t.\ \Delta_1^k+\ldots..+\Delta_n^k=\Delta \text{ e } \Delta_i^k\geq 0 \qquad \forall i$$

Per risolvere tale problema è possibile impiegare due approcci approssimativi. Il primo approccio consiste nel calcolo, fissato $m$ positivo intero e $\tau= b/m$ con $b$ l'incremento totale per tutte le simulazioni, di

$D_i\equiv$ EAPCS $(T^k_1, T^k_2,\ldots..., T^k_{i-1}, T^k_i+\tau,\ldots, T^k_n)$-APCS$(T^k_1, T^k_2,\ldots..., T^k_n)$.

A questo punto, si definisce l'insieme S(m) $\equiv$ {i:$D_i$ è all'interno degli $m$ più elevati} e si definisce $\Delta_i^k=\tau$ per tutte le $i$ appartenenti a S(m). In tal modo, si assegnano uguali incrementi a $m$ diversi sistemi.

Il secondo approccio, invece, impiega il metodo del gradiente per la definizione di lunghezze ulteriori di simulazione diverse per i diversi sistemi, usando la seguente formula approssimativa [39]:

$$\frac{\partial}{\partial N_i} EAPCS = \frac{APCS(N_1,\ldots.,N_i+\Delta,\ldots.,N_k)-APCS(N_1,\ldots..,N_i,..,N_k)}{\Delta}$$

con $\Delta$ sufficientemente piccolo.

Per quanto concerne la determinazione di $t_0$, in generale, un valore troppo ridotto incide sulla significatività delle stime di media e varianza, mentre un valore troppo elevato comporta spesso la non ottimizzazione del carico computazionale, portando ad un valore di molto superiore al limite P*. La figura mostra una relazione sperimentale tra il numero medio di replicazioni e $t_0$ per diversi valori di P*. Dal grafico si evince che il numero medio non varia sensibilmente in funzione di $t_0$ per P* superiori al 90%.

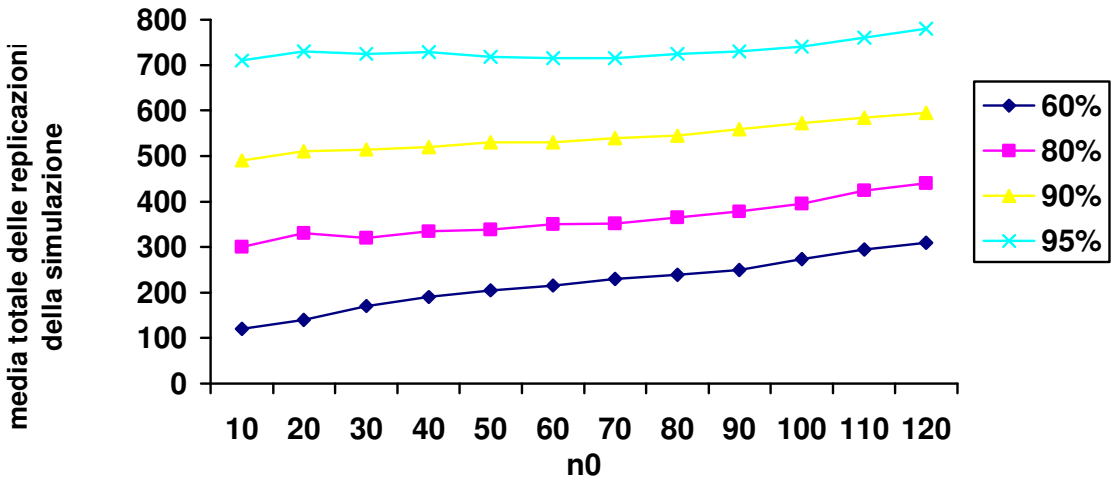



**Figura 4.7 Relazione tra il numero medio di replicazioni e $t_0$ per diversi valori di P*.**

Per quanto concerne la scelta del $\Delta$, invece, se questo risulta troppo ridotto il numero di iterazioni è piuttosto elevato, invece per valori alti di $\Delta$ il tempo di calcolo risulta eccessivamente ampio. In genere, è stato sperimentalmente dimostrato che valori compresi tra 15 e 30 sono adeguati nel caso di 10 designs. La figura mostra la relazione tra il $\Delta$ selezionato e il tempo totale, inteso come somma tra il tempo di simulazione e il tempo necessario alla risoluzione del problema di ottimizzazione, fissato P*=80%.

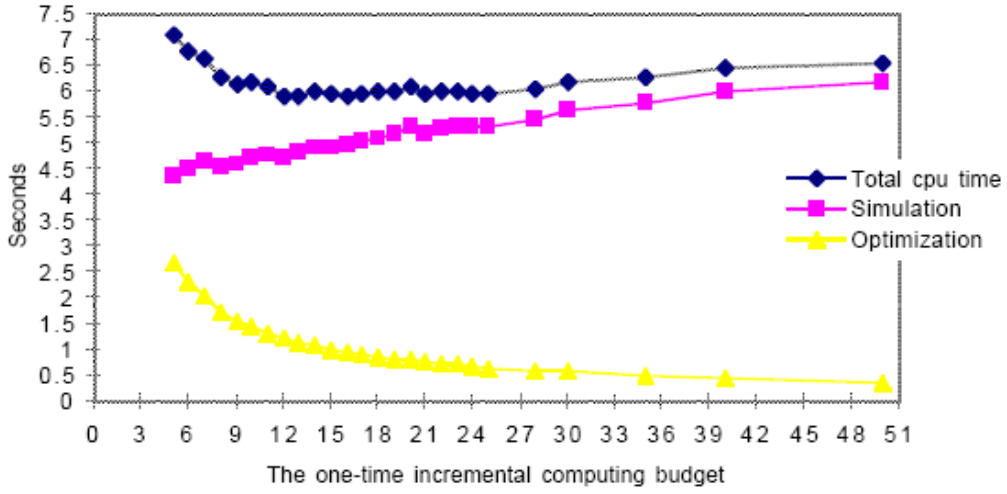



**Figura 4.8 Relazione tra D e il tempo totale computazionale fissato P***

La scelta di *m* è basata su valutazioni sperimentali, per le quali m piuttosto elevati sono adeguati nel caso di P* bassi e viceversa.

!Il confronto tra tale procedura e quella in due step effettuato sperimentalmente comporta che, in relazione ad un uso non solo della varianza ma anche della media nella determinazione del numero ottimale di replicazioni, tale algoritmo converge molto più velocemente rispetto alla procedura tradizionale.

Le tecniche analizzate fino a questo punto per il confronto di diversi sistemi comportano l'assunzione di un unico parametro di valutazione per la scelta della configurazione migliore. Tuttavia, gran parte dei sistemi reali presentano più criteri di valutazione. Seppur tali problemi siano sempre riconducibili a quelli con un unico parametro, valutando un unico indice ottenuto dai precedenti considerando i rispettivi pesi, sono state sviluppate diverse tecniche di analisi multi-obiettivo. In questo caso, non è possibile definire un'unica soluzione ottima o sub-ottima, ma un insieme di soluzioni non dominate. Il metodo proposto si basa sull'assunzione che le distribuzioni delle variabili siano continue, anche se è possibile generalizzare tale approccio anche al caso di variabili discrete, e che l'obiettivo sia la minimizzazione dei parametri. Per quanto concerne la definizione di soluzioni non dominate, relazionata al concetto di ottimalità di Pareto, si verifica che date due alternative con m misure di performance $\mu_i$: $\mu_{i1},\ldots,\mu_{im}$ e $\mu_j$: $\mu_{j1},\ldots,\mu_{jm}$ il design *j* –esimo domina quello *i-esimo*, definito come $\mu_j \prec \mu_i$, nel caso deteministico se sussiste tale condizione con almeno una disuguaglianza stretta:

$$\mu_{jk} \leq \mu_{ik} \qquad \text{per } k=1,2,\ldots,m.$$

Nel caso in cui si considerino variabili aleatorie, invece, si considera la probabilità che il *j–esimo* domini l' *i-esimo* secondo la seguente condizione con almeno una disuguaglianza stretta:

$$P(\mu_j \prec \mu_i)=P(\mu_{jk} \leq \mu_{ik} \quad \text{per } k=1,2..,m)$$

che risulta essere pari, nel caso di s-indipendenza dei parametri a:

$$P(\mu_j \prec \mu_i)=\prod^{m}_{k=1} P(\mu_{jk} \leq \mu_{ik})$$

A questo punto, è possibile considerare la probabilità cumulativa che il disegno *i-esimo* sia dominato dagli altri come:

$$\Psi_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} P(\mu_j \prec \mu_i)$$

$\psi_i$ è assunto come indice di performance e tanto più è prossimo allo zero tanto minore è la probabilità che il disegno *i-esimo* sia dominato dagli altri [109]. Si

definiscano $\tilde{F}_{i,k}$ e $\hat{F}_{i,k}$ rispettivamente come la variabile casuale associata alla media stimata per il *k-esimo* obiettivo e quella associata alla media dopo addizionali replicazioni per la configurazione *i-esima*, Anche in questo caso, si assume che la stima della distribuzione delle variabili nel caso di incremento di $\delta_0$ di una quantità $\delta_1$ possa essere espressa sulla base della distribuzione a $\delta_0$ come.

$$N(\ 1/\ \delta_0 \sum^{t0}_{j=1} X_{i,j},\ \sigma^2_i/\ \delta_0 + \delta_1).$$

Si può, quindi, definire l'indice di performance come:

$$\Psi_i = \sum_{j=1, j\neq i}^{n} \prod_{k=1}^{m} P(\tilde{F}_{j,k} \leq \tilde{F}_{i,k})$$

Fissato un valore ψ* predefinito, come indice di performance minimo richiesto per le configurazioni appartenenti all'insieme di Pareto $S_p$ di disegni non dominati, si stima ad ogni iterazione il valore di $\psi_i$ per ogni configurazione e si ordinano i valori ottenuti in senso crescente come segue:

$$\psi_{(1)} \leq \psi_{(2)} \leq \ldots \leq \psi_{(K)} \leq \ldots \leq \psi_{(n)}$$

Noto il valore di K, ovvero il numero di elementi dell'insieme di Pareto, tale che l'indice di performance è minore del valore limite ψ*, il problema è l'allocazione ottimale delle replicazioni per quelle configurazioni appartenente all'insieme degli elementi non dominati come segue:

$$\begin{aligned} & \min \sum_{i=1}^{n} \delta_i \\ \text{s.t.} \quad & \Psi_{(k)} \leq \Psi^* \qquad \forall k \leq K \\ & \delta_i \geq 0 \qquad \forall i \end{aligned}$$

In alternativa, è possibile minimizzare il più alto indice di performance vincolando il numero massimo di replicazioni come segue:

$$\min_{\delta_1,\delta_2,.....,\delta_n} \Psi_{(k)}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \delta_i \le N_{\max}$$

$$\delta_i \ge 0 \qquad \forall i$$

Al fine di risolvere tale problema si impiega una procedura iterativa. Tale algoritmo consiste nella realizzazione di $\delta_0$ replicazioni iniziali per ogni disegno, posto $N=n\ \delta_0$. A questo punto, si procede alla valutazione per ogni configurazione degli indici di performance $\psi_i$ e li si ordina in senso crescente $\psi_{(1)} \le \psi_{(2)} \le \ldots \le \psi_{(K)} \le \ldots \le \psi_{(n)}$. Si definisce l'insieme di Pareto, come quel dato insieme contenente i K indici più bassi. Se $\psi_{(K)} < \psi^*$ o $N>N_{max}$ l'algoritmo si arresta. Altrimenti, per ogni configurazione appartenente all'insieme di Pareto e per ogni disegno d=1,2,..,n, si considera la variazione dell'indice di performance $\Delta\psi_{id}$ in funzione delle replicazioni addizionali per la configurazione *d-esima*. Tale variazione può essere stimata come segue:

$$\Delta\Psi_{id} = \prod_{k=1}^{m} P(\tilde{F}_{d,k} \le \tilde{F}_{i,k}) - \prod_{k=1}^{m} P(\hat{F}_{d,k} \le \tilde{F}_{i,k}) \text{ se d}\neq\text{i}$$

$$\Delta\Psi_{id} = \sum_{j=1, j\neq i}^{n} \prod_{k=1}^{m} P(\tilde{F}_{j,k} \le \tilde{F}_{i,k}) - \sum_{j=1, j\neq i}^{n} \prod_{k=1}^{m} P(\tilde{F}_{j,k} \le \hat{F}_{i,k}) \text{ se d=i}$$

Quindi, è possibile calcolare la variazione totale dell'indice di performance come:

$$\Delta\Psi_d = \sum_{i\in S_p} \left|\Delta\Psi_{id}\right|$$

Si ordinano i disegni in ordine decrescente di $\Delta\Psi_d$ come $w_1,\ldots\ldots,w_n$ e si allocano le replicazioni addizionali $\delta_{wi}$, con i=1,2,…,p come segue:

$$\delta_{w_p} = \frac{\delta\Delta\Psi_{w_p}}{\sum_{i=1}^{p} \Delta\Psi_{w_p}}$$

e per i=1,2,…,p-1

$$\delta_{w_i} = \frac{\Delta\Psi_{w_i}}{\Delta\Psi_{w_p}} \delta_{w_p}$$

Si effettuano le ulteriori simulazioni e si pone

$$N=N+\sum_{i=1}^{p} \delta_{w_i}$$

e si ripete la valutazione. Tale procedura risulta molto efficace nel caso di impiego di più sistemi, l'unica limitazione attualmente presente è la determinazione a priori del numero di elementi K costituenti l'insieme di Pareto.

Esistono software sul mercato che aiutano nella determinazione della determinazione di k ed n. ASAP3 (Automated Simulation Analysis Procedure), ad esempio, oltre a calcolare *k* e *n*, costruisce l'intervallo di confidenza che soddisfa la precisione assoluta o relativa richiesta. L'ASAP3 richiede i seguenti input:

- un processo di output $\{X_j, j = 1,2,...,m\}$ generato da un singolo run di una simulazione di lunghezza m;
- il livello di confidenza $1-\alpha$ desiderato per l'intervallo di confidenza, con $0 < \alpha < 1$;
- la precisione assoluta o relativa richiesta.

L'ASAP3 fornisce in uscita:

- la stima di $\nu$;
- l'intervallo di confidenza per $\nu$ al $100(1-\alpha)\%$ che soddisfa la precisione relativa o assoluta specificata

oppure una dimensione *k'* dei batches più grande, nel caso in cui i dati iniziali non siano sufficientemente elevati da consentire il calcolo dell'intervallo di confidenza [45, 58].

*Il metodo rigenerativo*

Questo metodo si basa sull'identificazione di un indice di tempo in corrispondenza del quale il processo $\{X_i\}$ probabilmente riparte e utilizza queste epoche di rigenerazione al fine di ottenere variabili aleatorie I.I.D. che consentano la stima della media $\nu$. Più precisamente, questo metodo assume che ci siano indici di tempo casuali $1 \le T_1 < T_2 < \ldots$ tali che la porzione $\{X_{T_i} + j, j \ge 0\}$ abbia la stessa distribuzione per ogni i e sia indipendente dalla porzione che precede $T_i$. La porzione del processo compresa tra due epoche rigenerative successive è detta ciclo.

Definite le v.a.

$$Y_i = \sum_{j=T_i}^{T_{i+1}-1} X_j \qquad \text{per } i = 1,2,\ldots$$
$$Z_i = T_{i+1} - T_i$$

la media $\nu$ è data da

$$\nu = \frac{E(Y)}{E(Z)}.$$

Inoltre l'intervallo di confidenza per $\nu$ può essere costruito usando le variabili casuali $Y_i - \nu Z_i, i = 1,2,\ldots$ e il teorema del limite centrale.

Il metodo rigenerativo è difficile da applicare nella pratica perché la maggioranza dei sistemi reali non presenta punti di rigenerazione oppure, se li ha, la lunghezza dei cicli è molto grande [31].

*Il metodo dell'analisi spettrale*

Questo metodo ipotizza che il processo stocastico tempo discreto $Y_1, Y_2, \ldots$ sia a covarianza stazionaria con valore atteso $E(Y_i) = \nu$. Un processo stocastico tempo discreto $Y_1, Y_2, \ldots$ è detto a covarianza stazionaria se

$$\mu_i = \mu$$
$$\sigma_i^2 = \sigma^2$$

per $i = 1,2,\ldots$, $-\infty < \mu < \infty$, $\sigma^2 < \infty$, e $C_{i,i+j} = Cov(Y_i, Y_{i+j})$ è indipendente da i per $j = 1,2,\ldots$. Quindi la covarianza tra due osservazioni $Y_i$ e $Y_{i+j}$ dipende solo da j e non dai valori attuali di tempo i e i+j.

La covarianza $C_j$ può essere stimata come segue

$$\hat{C}_j = \frac{\sum_{i=1}^{n-j}\left[Y_i - \overline{Y}(n)\right]\left[Y_{i+j} - \overline{Y}(n)\right]}{n-j}.$$

Sotto questa ipotesi è possibile mostrare che uno stimatore della varianza di $\overline{Y}(m)$ è dato dalla relazione

$$Var\left[\overline{Y}(m)\right] = \frac{C_0 + 2\sum_{j=1}^{m-1}(1 - j/m)C_j}{m} \qquad \text{(e 4.10)}$$

Il nome di questo metodo si basa sul fatto che, per $m \to \infty$,

$$m \cdot Var\left[\overline{Y}(m)\right] \to 2\pi g(0)$$

dove $g(\tau)$ è detto *spettro* del processo con frequenza $\tau$, ed è definito dalla trasformata di Fourier $g(\tau) = (2\pi)^{-1}\sum_{j=-\infty}^{\infty} C_j \exp(-i\tau j)$, per $|\tau| \le \pi$ e $i = \sqrt{-1}$. Dunque, per $m$ grande, $Var\left[\overline{Y}(m)\right] \approx \frac{2\pi g(0)}{m}$ e il problema della stima di $Var\left[\overline{Y}(m)\right]$ può essere ricondotto alla stima dello spettro a frequenza nulla.

Comunque, la stima della varianza si calcola facilmente dalla (e 4.10) sostituendo a $C_j$ la sua stima $\hat{C}_j$ [10].

*Metodo delle serie temporali standardizzate*

Si supponga che il processo $Y_1, Y_2, \ldots$ sia strettamente stazionario con media $E(Y_i) = \nu$. Si definisce strettamente stazionaria una distribuzione di v.a. $Y_{i_1+j}, Y_{i_2+j}, \ldots, Y_{i_n+j}$ indipendenti da j per ogni indice di tempo $i_1, i_2, \ldots, i_n$. Si supponga di eseguire un run di una simulazione di lunghezza m e di dividere le osservazioni $Y_1, Y_2, \ldots, Y_m$ in n gruppi (batches) di dimensione k. Sia $\overline{Y}_j(k)$ la media campionaria delle k osservazioni del j-esimo batch. La media campionaria $\overline{Y}(m)$ calcolata su tutti i batches è uno stimatore di $\nu$; se m è grande, $\overline{Y}(m)$ sarà approssimativamente distribuita normalmente con media $\nu$ e varianza $\tau^2/m$, dove $\tau^2 = \lim_{m\to\infty} m \cdot Var\left[\overline{Y}(m)\right]$.

Inoltre, per un fissato numero di batches n e per $k \to \infty$, sia

$$A = \left( \frac{12}{k^3 - k} \right) \sum_{j=1}^{n} \left\{ \sum_{s=1}^{k} \sum_{i=1}^{s} \left[ \overline{Y}_j(k) - Y_{i+(j-1)k} \right] \right\}^2$$

una v.a. distribuita come una chi-quadro con n gradi di libertà e indipendente da $\overline{Y}(m)$. Dunque per $k \to \infty$ è possibile trattare

$$\frac{\left[ \overline{Y}(m) - \nu \right] / \sqrt{\tau^2 / m}}{\sqrt{\left( A / \tau^2 \right) / n}} = \frac{\overline{Y}(m) - \nu}{\sqrt{A / mn}}$$

come una v.a. t-Student con n gradi di libertà, e un intervallo di confidenza per $\nu$ approssimato al $100(1-\alpha)\%$ è dato da

$$\overline{Y}(m) \pm t_{n,1-\frac{\alpha}{2}} \sqrt{A / mn} \, .$$

*Il metodo autoregressivo*

Sia $Y_1, Y_2, \ldots$ un processo a covarianza stazionaria con media $E(Y_i) = \nu$. Esso può essere rappresentato con un modello autoregressivo di ordine p:

$$\sum_{j=0}^{p} b_j \left( Y_{i-j} - \nu \right) = \varepsilon_i$$

dove $b_0 = 1$ e $\{\varepsilon_i\}$ è una sequenza di variabili casuali non correlate con media $\mu_\varepsilon = 0$ e varianza $\sigma_\varepsilon^2$.

È possibile mostrare che, per $m \to \infty$,

$$m \cdot Var\left[ \overline{Y}(m) \right] \to \frac{\sigma_\varepsilon^2}{\left( \sum_{j=0}^{p} b_j \right)^2} .$$

Dunque, per m grande, la stima della varianza e l'intervallo di confidenza per $\nu$ approssimato al $100(1-\alpha)\%$ sono dati da

$$Var\left[ \overline{Y}(m) \right] = \frac{\hat{\sigma}_\varepsilon^2}{m \left( \hat{b} \right)^2}$$

$$\overline{Y}(m) \pm t_{\hat{f},1-\frac{\alpha}{2}} \sqrt{Var\left[ \overline{Y}(m) \right]}$$

con $\hat{b} = 1 + \sum_{j=1}^{\hat{p}} \hat{b}_j$, $\hat{p}$ e $\hat{\sigma}_\varepsilon^2$ stime rispettivamente dell'ordine p e della varianza $\sigma_\varepsilon^2$, mentre una stima dei gradi di libertà $\hat{f}$ è fornita dalla relazione

$$\hat{f} = \frac{m\hat{b}}{2\sum_{j=0}^{\hat{p}} (\hat{p} - 2j)\hat{b}_j}$$

### *4.3.3. Stima di altre misure di prestazione*

Analogamente a quanto visto per le simulazioni con terminazione, si supponga di voler stimare la probabilità stazionaria

$$p = \Pr\{Y \in B\}$$

con $B$ insieme di numeri reali.
Sia $Z$ una v.a. stazionaria definita come:

$$Z = \begin{cases} 1 & se \quad Y \in B \\ 0 & altrimenti \end{cases}.$$

Allora:

$$p = \Pr\{Y \in B\} = \Pr\{Z = 1\} = 1 \cdot \Pr\{Z = 1\} + 0 \cdot \Pr\{Z = 0\} = E(Z)$$

cioè la stima di $p$ è equivalente alla stima della media stazionaria $E(Z)$. In particolare, sia:

$$Z_i = \begin{cases} 1 & se \quad Y_i \in B \\ 0 & altrimenti \end{cases}$$

dove $Y_1, Y_2, \ldots$, per $i = 1,2,\ldots$, è il processo stocastico di interesse. Allora è possibile, ad esempio, applicare il metodo delle repliche/cancellazioni al processo di output $Z_1, Z_2, \ldots$ ed ottenere così la stima e l'intervallo di confidenza per $E(Z) = p$. Bisogna notare, però, che il periodo di warm-up per il processo binomiale $Z_1, Z_2, \ldots$ può essere diverso da quello del processo originario $Y_1, Y_2, \ldots$, dunque deve essere ricalcolato.
Per le simulazioni senza terminazione la stima dei quantili risulta molto difficile dal punto di vista concettuale e computazionale, ossia in termini di numero di osservazioni necessarie ad ottenere la precisione desiderata per la stima. La

procedura di stima dei quantili si basa sull'algoritmo $P^2$ che ha il vantaggio di non richiedere un numero molto elevato di osservazioni. Esso è una procedura sequenziale basata su un metodo spettrale, comunque implementabile con non poche difficoltà.

## 4.4. Analisi statistica dei parametri ciclici

Si consideri il processo stocastico $Y_1, Y_2, \ldots$ generato da una simulazione senza terminazione che non ha una distribuzione stazionaria. Si supponga di dividere l'asse dei tempi in intervalli di tempo contigui e di uguale lunghezza, detti *cicli.* Sia $Y_i^C$ una v.a. definita sull'*i*-esimo ciclo e si supponga che le variabili $Y_1^C, Y_2^C, \ldots$ siano confrontabili. Si supponga inoltre che il processo $Y_1^C, Y_2^C, \ldots$ abbia una distribuzione stazionaria $F^C(y) = \Pr\{Y^C \le y\}$, cioè indipendente dal ciclo. Allora una misura di prestazione si definisce *parametro ciclico stazionario* se è una caratteristica di $Y^C$, come la media $E(Y^C) = \nu^C$.

Si tenga presente che se la definizione dei cicli non è appropriata, il corrispondente processo $Y_1^C, Y_2^C, \ldots$ non ha una distribuzione stazionaria. Ciò accade quando i parametri del modello di simulazione cambiano continuamente nel tempo, ad esempio il tasso di arrivo delle parti all'interno del sistema o le regole di carico. In questi casi è comunque possibile fissare degli orizzonti temporali nei quali questi parametri si mantengono costanti; ciò equivale a fissare un evento $E$ in corrispondenza del quale termina la simulazione e quindi risultano appropriate le tecniche di analisi per simulazioni con terminazione.

La stima di un parametro ciclico stazionario non è altro che un caso particolare della stima di un parametro stazionario, di conseguenza è possibile utilizzare per la stima tutte le tecniche viste per le simulazioni senza terminazione.

## 4.5. Misure multiple di prestazione

Molto spesso, nella simulazione di sistemi reali, si è interessati a stimare un certo numero di misure di prestazione, importanti per la valutazione del comportamento del sistema. Utilizzando i risultati di un singolo run di una

simulazione è possibile stimare tutti i parametri di interesse, che sono tra loro correlati.

Siano $\mu_s$ le misure di prestazione e $I_s$ i relativi intervalli di confidenza al $100 \cdot (1-\alpha_s)\%$, con $s = 1,2,...,k$. La probabilità che tutti i $k$ intervalli di confidenza contengono simultaneamente le rispettive misure vere soddisfa la seguente disuguaglianza

$$\Pr\left\{\cap_{S=1}^{K} (\mu_s \in I_s)\right\} \geq 1 - \sum_{s=1}^{k} \alpha_s$$

nota come *disuguaglianza di Bonferroni*. Ad esempio, se si vogliono costruire $k$=10 intervalli di confidenza al 99%, con $\alpha_s = 0,01$ e con $s = 1,2,...,10$, allora il livello di confidenza generale sarà almeno pari al 90%, ovvero

$$\Pr\{\mu_s \in I_s, s = 1,2,...,10\} \geq 1 - 10 \cdot \alpha_s = 0.90 .$$

Naturalmente l'utilizzo del *Metodo di Bonferroni* per la stima delle misure multiple di prestazione presuppone la scelta di un metodo per il calcolo degli intervalli di confidenza delle singole misure di prestazione [21, 22].

# 5.La tecnica Bootstrap nella Simulazione

## 5.1. Stima degli intervalli di confidenza mediante Bootstrap

L'impiego del metodo Bootstrap nell'ambito della simulazione dei processi produttivi risulta particolarmente vantaggioso, in relazione al fatto che tali processi presentano spesso output con distribuzione non nota a priori. In tal caso, sarebbe necessario, al fine di impiegare le tradizionali tecniche di valutazione dei risultati, generare un campione di dati molto ampio che permetta di supporre la validità del teorema del limite centrale. Seppur tale approccio possa risultare in prima analisi di semplice realizzazione, è necessario considerare che l'onere computazionale dovuto alla generazione di un campione piuttosto ampio, in termini di tempi di realizzazione della simulazione, risulta spesso eccessivo. Per questo motivo la tecnica bootstrap risulta particolarmente vantaggiosa soprattutto in due delle fasi dello sviluppo della simulazione del sistema. Tale metodologia, infatti, può essere impiegata sia nell'ambito della validazione dei modelli di simulazione, sia durante l'analisi dei risultati. In quest'ultimo caso, in genere, si suppone che gli input della simulazione abbiano distribuzione nota. Tuttavia, il metodo Bootstrap non-parametrico può anche essere impiegato ipotizzando di non conoscere la distribuzione dei dati in input, ma di possedere solo un campione di dimensioni finite, che può essere impiegato al fine di effettuare la stima o la costruzione delle distribuzioni in input. In questo caso, si considera il vettore degli output come funzione del vettore $\mathbf{X}$ degli input, ovvero $\mathbf{Y}(\mathbf{X})=t(\mathbf{X})$. Ne consegue che i campioni Bootstrap $\mathbf{X^*_k}$ sono ottenuti a partire dal campione $\mathbf{X}$ originario. Utilizzando talii campioni come input della simulazione, si ottiene per ciascuna replicazione il vettore $\mathbf{Y}(\mathbf{X_k^*})=t(\mathbf{X_k^*})$[41].

Per analizzare l'utilità dell'impiego del metodo Bootstrap, è necessario considerare che l'inferenza statistica è basata sull'analisi della distribuzione campionaria di statistiche predefinite, al fine di ricavare informazioni sulla popolazione di partenza. Il metodo Bootstrap permette di stimare la distribuzione campionaria, anche a partire da una distribuzione della popolazione non nota, con un campione di dimensioni ridotte, ovvero laddove non sia possibile ricorrere al teorema del limite centrale. Tale metodo non è equivalente all'acquisizione di

ulteriori dati per incrementare l'accuratezza della stima. Al contrario, si impiegano i dati Bootstrap ottenuti per capire come varia la statistica ottenuta dal campione di partenza a causa dell'errore generato dal campionamento casuale. In primo luogo, si costituiscono i campioni Bootstrap attraverso il campionamento con rimessa dal campione di partenza. Ogni campione, ottenuto con rimessa, ha la stessa dimensione di quello di partenza. Calcolato lo stimatore per ogni campione Bootstrap, la distribuzione empirica delle statistiche, definita come *distribuzione Bootstrap*, permette di ottenere informazioni sulla forma, centro e varianza della distribuzione campionaria della statistica di partenza. La Figura 5.1 mostra la differenza tra l'approccio tradizionale e quello basato sull'impiego del Bootstrap. Nel primo caso, al fine di ottenere una stima del parametro incognito, si effettuano più campionamenti, per la simulazione più replicazioni, e si valuta la distribuzione delle statistiche campionarie. L'impiego del Bootstrap, al contrario, permette di sviluppare intervalli di confidenza per i parametri di interesse a partire da un unico campione originario [76].

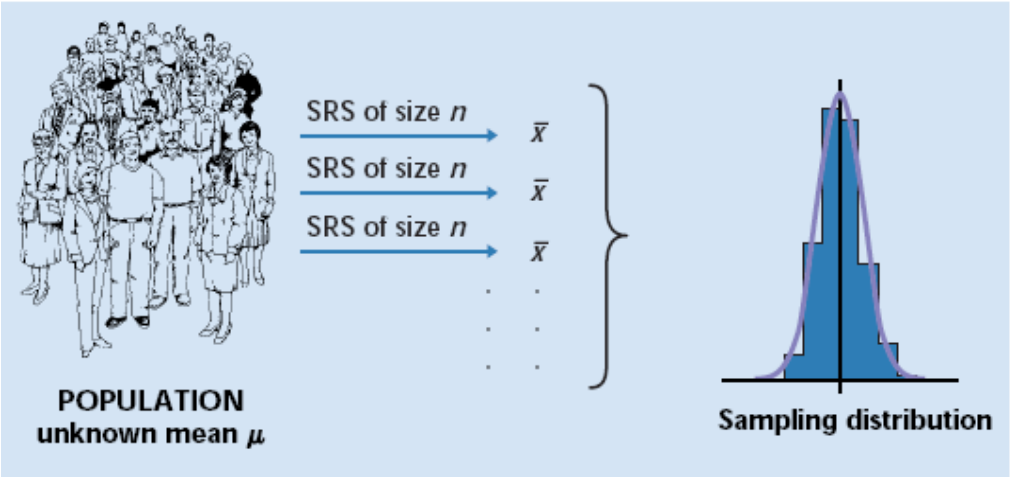



**Figura 5.1 (a) Tecnica tradizionale di stima dei parametri**

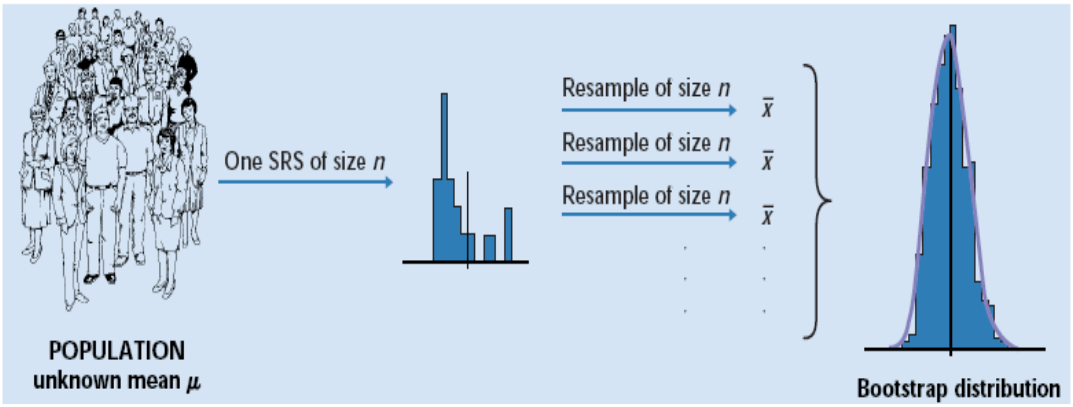



**Figura 5.1 (b) Stima dei parametri attraverso l'impiego della distribuzione Bootstrap**

Si ipotizzi che il parametro di interesse θ sia stimato a partire da un campione di dimensione $n$ **Y**=($Y_1$,$Y_2$,.....,$Y_n$) attraverso l'impiego della seguente statistica:

$$\hat{\theta} = t(Y_1, Y_2, ...., Y_n)$$

con distribuzione t(**Y**) non nota, il che implica che non è possibile considerare attraverso tale relazione l'errore di stima.

Il metodo tradizionale prevede la generazione di $B$ campioni $\mathbf{Y_b}$ a partire da una distribuzione della popolazione F(.), sulla base dei quali calcolare le $B$ statistiche ordinate { $\hat{\theta}^{(1)}$,..... $\hat{\theta}^{(B)}$} come mostrato in Figura 5.2.

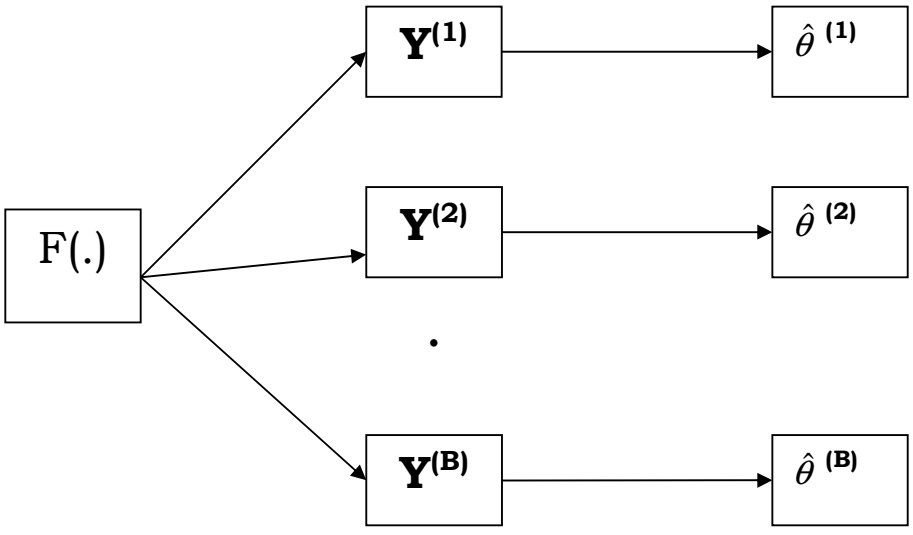



**Figura 5.2**

Sulla base di tali statistiche, è possibile stimare la distribuzione campionaria che converge a quella vera per *B* che tende all'infinito. In particolare tale risultato si ottiene solo se la dimensione di ciascun campione Bootstrap è uguale a quella del campione originario. Alcuni studi, tra i quali quelli di Bickel e Freedman [24] sono stati effettuati per impiegare campioni Bootstrap di dimensioni differenti rispetto a quella del campione di partenza ad esempio per la stima dell'errore standard. Tuttavia, queste procedure non sembrano condurre a nessun vantaggio operativo. Per questo motivo, nella trattazione saranno sempre considerati campioni Bootstrap con stessa ampiezza di quelli di partenza.[61] Tuttavia, questo metodo richiede la realizzazione di *B* replicazioni della simulazione se a partire da ciascuna di esse è possibile ottenere *n* dati, oppure *n*B* replicazioni nel caso in cui la simulazione generi un unico output, eventualità spesso riscontrata nell'ambito della terminating simulation di sistemi produttivi.
Il principio Bootstrap, invece, permette di ottenere informazioni in merito alla relazione tra θ e la variabile casuale $\hat{\theta}(\mathbf{Y})$ attraverso l'analisi della connessione tra $\hat{\theta}(\mathbf{Y_{obs}})$ e $\hat{\theta}(\mathbf{Y^*})$, dove **Y*** è ottenuto a partire dal campione di partenza $\mathbf{Y_{obs}}$.. Il principio basilare su cui si basa il metodo Bootstrap, in altri termini, è la realizzazione di *B* campioni Bootstrap a partire da un unico campione originario $\mathbf{Y_{obs}}=(Y_1,Y_2,\ldots\ldots,Y_n)$ di dimensione *n*:

$$\boldsymbol{Y}_b^* = \left(Y^*_1, Y^*_2, \ldots\ldots, Y^*_n\right) \qquad \text{con b=1,2,...B}$$

Lo stimatore del parametro ottenuto da ciascun campione Bootstrap è:

$$\hat{\theta}_b^* = t(Y_b^*)$$

La stima della distribuzione vera dello stimatore $\hat{\theta}$ si ottiene a partire dall'insieme ordinato $\{\hat{\theta}^*(1), \hat{\theta}^*(2), \ldots., \hat{\theta}^*(B)\}$. In questo caso, i campioni Bootstrap non sono ottenuti a partire dalla distribuzione vera della popolazione di partenza F(.), ma dalla migliore stima di questa ottenuta dal campione di partenza Fm(.)come mostrato in Figura 5.3.

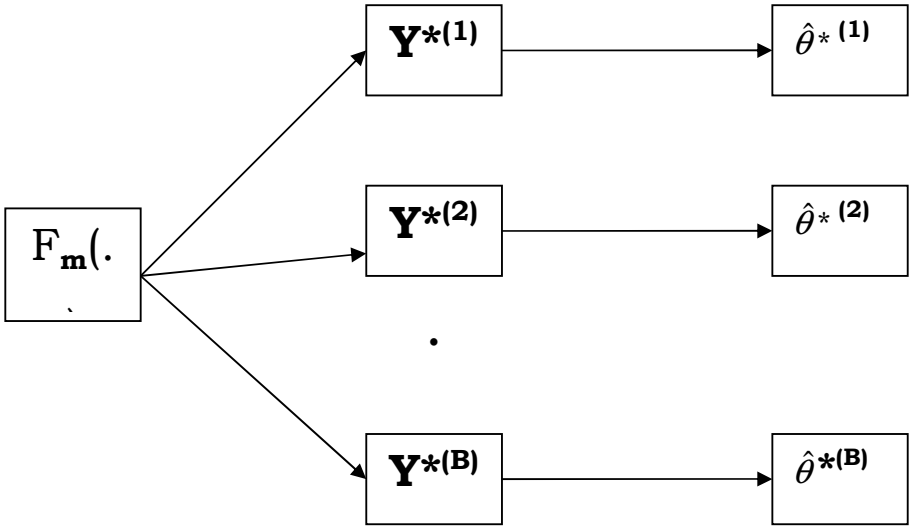



**Figura 5.3**

Tale metodologia, quindi, si basa sull'ipotesi che Fm(.) converga ad F(.). [43] Poiché Yobs è un campione casuale, tale ipotesi è garantita dal teorema di Glivenko-Cantelli, che assicura la convergenza uniforme con probabilità unitaria. Infatti, si definisca

$$\hat{F}_n(x) := \frac{1}{n}.I\{1 \le i \le n \mid Y_i \le y\}$$

come una funzione di distribuzione empirica che approssima la funzione incognita F, con I funzione di conteggio degli elementi dell'insieme. La massima deviazione della distribuzione empirica dalla variabile casuale che ne sta alla base può essere considerata pari a:

$$d_n = \sup_y |\hat{F}_n(y) - F(y)|$$

Allora la differenza dn converge a zero con probabilità unitaria, ovvero:

$$P(\lim_{n\to\infty} d_n = 0) = 1$$

Inoltre, preso un campione di dimensione *n* la probabilità che un particolare valore $y_i$ non sia compreso nel singolo ricampionamento è:

$$\Pr(Y_j^* \neq Y_i, 1 \leq j \leq n) = \left(1 - \frac{1}{n}\right)^n$$

nell'ipotesi semplificativa che tutti gli elementi del campione originario siano distinti. Di conseguenza, la frazione attesa di elementi non compresi nel singolo campione Bootstrap è $(1-(1/n))^n$. Normalmente, quindi, circa il 37% di elementi non sono considerati in ciascun ricampionamento per tutti i valori di *n*. Se ne deduce che il Bootstrap può essere impiegato per tutte le dimensioni campionarie.[148]

Il campione **Y*** può essere ottenuto sia attraverso il campionamento con reimmissione dall'insieme $\mathbf{Y_{obs}}$, sia dal campionamento dalla distribuzione parametrizzata attraverso $\hat{\theta}(\mathbf{Y_{obs}})$. La prima tecnica è definita come metodo *Bootstrap nonparametrico*, la seconda come *Bootstrap parametrico.*

Il metodo di ricampionamento non-parametrico, permette di non formulare nessuna ipotesi sul modello o distribuzione dei dati. Considerato, quindi, il vettore $\mathbf{Y_{obs}}$ costituito da *n* osservazioni indipendenti, è possibile effettuare il ricampionamento con reimmissione al fine di ottenere il vettore Bootstrap **Y*** di dimensione *n*. Sulla base di tale campione Bootstrap, è possibile stimare $\hat{\theta}^* = \hat{\theta}($ **Y*** ). Questa valutazione è effettuata per *B* campioni Bootstrap. Il metodo parametrico, invece, considera un modello $F_m(\mathbf{Y}, \eta)$ noto, che risulta funzione dei parametri η non determinati. In questo caso, i campioni Bootstrap sono ottenuti a partire dalla funzione $F_m(\mathbf{Y}, \hat{\eta})$ dove $\hat{\eta}$ è, in genere, lo stimatore di massima verosimiglianza ottenuto a partire dal campione originario $\mathbf{Y_{obs}}$. Anche in questo caso, si considerano *B* campioni Bootstrap a partire dai quali si valuta lo stimatore $\hat{\theta}_b^* = \hat{\theta}($ $\mathbf{Y_b^*}$ ) con b=1,2,...,B. Le due metodologie si basano su assunzioni completamente differenti. Risulta, quindi, fondamentale la scelta della tecnica da impiegare in modo che il processo simulato sia il più possibile rappresentativo del sistema reale. In relazione alla difficoltà di determinare il modello di distribuzione degli output di simulazione per sistemi molto complessi, soprattutto nel caso di input con differenti distribuzioni, in genere risulta maggiormente corretto l'impiego di un metodo non parametrico. Per questo motivo, soprattutto nell'ambito della definizione e valutazione degli intervalli di confidenza, si discuterà con un maggior livello di dettaglio tale procedura [33].

In relazione alle applicazioni Bootstrap inerenti all'analisi dei risultati, sotto l'ipotesi di distribuzione degli input nota, è possibile valutare l'accuratezza dello stimatore in termini di bias, varianza e intervalli di confidenza. In particolare, la distribuzione Bootstrap ha una forma e ampiezza che approssima la distribuzione dello stimatore. Tuttavia, la distribuzione Bootstrap non è centrata sul valore vero, ma su quello dello stimatore ottenuto dal campione di partenza. In primo luogo, la distribuzione Bootstrap può essere impiegata al fine di verificare se lo stimatore impiegato è *biased*, ovvero se la sua distribuzione non è incentrata sul valore vero. In particolare, il bias è pari alla differenza del valore vero e la media della distribuzione campionaria, ovvero:

$$bias = E(\hat{\theta}) - \theta .$$

Poiché il valore vero risulta incognito è possibile calcolare il bias a partire dalla distribuzione Bootstrap. Infatti, pur presentando un diverso centro, le due distribuzioni presentano bias molto prossimi. In altri termini, calcolando il bias della distribuzione bootstrap è possibile risalire al bias della statistica di interesse. Nel caso del bias, definito $\overline{\theta}* = \frac{1}{B}\sum_{b=1}^{B}\theta_b *$ , si ha che:

$$\hat{b}ias* = \overline{\theta} * - \hat{\theta}$$

Come regola generale, Efron e Thibshirami [61] affermano che un bias inferiore del 25% dell'errore standard può essere ignorato. Laddove, invece, il bias risulti consistente è possbile considerare una stima corretta dal bias pari a [74]:

$$\hat{\theta} - \hat{b}ias* = \hat{\theta} - (\overline{\theta} * - \hat{\theta}) = 2\hat{\theta} - \overline{\theta} *$$

Allo stesso modo, è possibile stimare la deviazione standard della statistica di interesse attraverso la valutazione dell'errore standard della distiribuzione Bootstrap pari a:

$$\hat{S}E* = \frac{1}{B}\sum_{b=1}^{B}(\theta *_b - \overline{\theta}*)$$

La deviazione Bootstrap risulta essere per gran parte delle statistiche $\hat{\theta}$ un buon stimatore del valore vero della deviazione standard $SE(\hat{\theta})$.

Il metodo più diffuso finalizzato alla valutazione dell'intervallo di confidenza per una data statistica d'interesse $\hat{\theta}$ prevede lo sviluppo di un intervallo standard che può essere espresso come:

$$\hat{\theta} \pm z^{(\alpha)} \hat{\sigma}$$

Il termine $\hat{\sigma}$ indica la stima della deviazione standard della statistica e $z^{(\alpha)}$ è il 100α-esimo percentile di una variabile aleatoria Normale standard [54]. In altri termini, l'intervallo standard si fonda sull'assunzione che:

$$\hat{\theta} \sim N(\theta, \sigma^2)$$

In realtà, anche nel caso di distribuzioni non perfettamente Normali tale metodo è impiegato sulla base dell'ipotesi di validità del teorema del limite centrale. Tuttavia, soprattutto in questi casi, si presuppone un'approssimazione asintotica che comporta, in genere, un elevato scarto rispetto all'intervallo esatto di confidenza, laddove ne sia possibile il calcolo. Le stesse problematiche si presentano nel caso di deviazione standard incognita, ovvero laddove si consideri come funzione ancillare la t-Student. Per questo motivo, sono state sviluppate delle tecniche basate sul metodo Bootstrap per il calcolo dell'intervallo di confidenza con una maggiore accuratezza rispetto al metodo tradizionale. In particolare, tali tecniche permettono di prescindere dall'ipotesi di Normalità della distribuzione. Si verifica, quindi, la possibilità di determinare gli intervalli di confidenza per i parametri di interesse anche se la distribuzione risulta incognita o particolarmente asimmetrica e skewed.

Il confronto tra i metodi tradizionali e quelli Bootstrap permette di affermare che questi ultimi risultano, non solo asintoticamente più accurati ma anche più corretti. L'accuratezza si riferisce all'errore di copertura. In particolare, è necessario considerare la copertura per ciascuna coda. In altri termini, spesso si può verificare che gli intervalli standard presentino la stessa copertura totale di quelli Bootstrap, essendo caratterizzati da un errore in eccesso su una coda e uno in difetto sull'altra. Per questo motivo, l'analisi di accuratezza è effettuata considerando un intervallo di confidenza ad una coda. Con errore di prima specie pari ad α tale intervallo presenta, nel caso di accuratezza del primo ordine, come per gli intervalli standard, una copertura effettiva pari ad $\alpha+O(1/n^{1/2})$. Gli

intervalli Bootstrap, invece, presentano un'accuratezza del secondo ordine, caratterizzata da una probabilità di copertura pari ad α+O(1/n), ovvero

$$\Pr(\theta \leq \hat{\theta}[\alpha]) = \alpha + O(1/n).$$

Allo stesso modo, tali intervalli rappresentano una correttezza del secondo ordine. In altri termini, i limiti si discostano da quelli esatti per un $O(n^{-3/2})$. Tale tipo di correttezza e accuratezza, comunque, non permettono di ottenere intervalli esatti. In particolare, gli intervalli non parametrici presentano un errore di terza specie connesso alla valutazione della varianza considerando *n* e non *n-1* gradi di libertà. In ogni caso, tali intervalli risultano più efficienti di quelli standard [60].Infine, un altro elemento da considerare a parità di copertura è la lunghezza, intesa come la distanza tra i due estremi, e la forma definite rispettivamente come:

$$l = \hat{\theta}[1-\alpha] - \hat{\theta}[\alpha]$$

e

$$f = (\hat{\theta}[1-\alpha] - \hat{\theta}) / (\hat{\theta} - \hat{\theta}[\alpha]).$$

In particolare, nel caso di intervalli standard, la forma risulta essere sempre pari all'unità In tal senso, se la distribuzione è skewed verso destra, l'intervallo standard presenta una stima poco ottimistica del limite inferiore e troppo di quello superiore.

Per quanto concerne i metodi Bootstrap, saranno analizzate le seguenti tecniche:

- Metodo **Bootstrap-t**.
- Metodo **Bias corrected and accelereted** (BCa);

Per quanto concerne il metodo *Bootstrap-t*, tale procedura consiste nella definizione della variabile $T = \dfrac{\hat{\theta} - \theta}{\hat{\sigma}}$ .

A partire da tale statistica, si considera il valore $T^{(\alpha)}$, ovvero il 100α-esimo percentile della distribuzione di *T*. L'intervallo di confidenza ad una coda che si ottiene, risulta essere pari a:

$$(-\infty, \hat{\theta} - \hat{\sigma} T^{*(\alpha)})$$

con

$$T^* = \frac{\hat{\theta}^* - \hat{\theta}}{\hat{\sigma}^*},$$

In altri termini, si considera per ogni campione Bootstrap il valore $T_b^*$. In questo modo, è possibile considerare la distribuzione empirica di *T* a partire dal

campione ordinato {$T^*_{(1)}$,……,$T^*_{(B)}$}. Si ottiene, quindi, il valore $T^*_{(\alpha)}$ come $T^*_{\lfloor B\alpha \rfloor}$. Nel caso della media, in generale, risulta che:

$$\hat{\theta} = \sum_{i=1}^{n} \frac{y_i}{n}$$

e $$\hat{\sigma}^2 = \frac{1}{(n-1)} \sum_{i=1}^{n} (y_i - \hat{\theta})^2$$

In genere, tale metodo è impiegato laddove il parametro da stimare sia la media. Infatti, per statistiche più complessa risulta difficoltoso stimare la varianza. Infatti, tale valutazione può essere effettuata anche in casi più complessi, ma avvalendosi di numerose approssimazioni che causano una perdita in termini di esattezza dell'intervallo di confidenza. In alternativa, è possibile realizzare un Bootstrap di secondo livello finalizzato alla stima della deviazione standard. Per ogni campione ottenuto $\mathbf{Y^*_b}$ si considerano *M* campioni Bootstrap $\mathbf{Y^*_m}$ , in modo da ottenere $\hat{\theta}_m{}^{**}=\hat{\theta}(\mathbf{Y_m^{**}})$. Ne consegue che la varianza di $\hat{\theta}(\mathbf{Y_b^*})$ può essere stimata come:

$$\hat{\sigma}^{*2} = \frac{1}{M-1} \sum_{j=1}^{M} (\hat{\theta}_j^{**} - \hat{\theta}(y_b^*))^2$$

Reiterando tale procedura per tutti i *B* campioni Bootstrap, si ottiene la stima della varianza di $\hat{\theta}(\mathbf{Y_{obs}})$ come:

$$\hat{\sigma}^2 = \frac{1}{B-1} \sum_{i=1}^{B} (\hat{\theta}_i^* - \hat{\theta}(y_{obs}))^2 .$$

Considerato che tale procedura richiede la generazione di *B*M* campioni Bootstrap, con *M* in genere almeno pari a 25, si evince che l'onere computazionale risulta essere piuttosto elevato. Inoltre, la stima della varianza è effettuata ipotizzando una indipendenza tra $\hat{\sigma}$ e $\hat{\theta}$. Tale ipotesi può essere verificata considerando il grafico di $\hat{\sigma}^*$ in funzione di $\hat{\theta}^*$. Se da questa analisi preliminare si evince una correlazione, allora è necessario considerare, attraverso una tecnica di regressione non lineare, la stima della funzione *s*, sulla base dalle coppie $(\hat{\theta}^*, \hat{\sigma}^*)$, tale che:

$$\hat{\sigma}^* = s(\hat{\theta}^*)$$

A partire da tale stima, si assume, come mostrato dallo sviluppo in serie di Taylor, che $V(x)=\int^{x} 1/s(x)d\theta$ è una funzione approssimata di stabilizzazione della varianza. A questo punto, è possibile riportare un secondo insieme di *B* stimatori $\hat{\theta}^*$ sulla scala stabilizzata, calcolare un intervallo di confidenza per V(θ) e, attraverso la funzione inversa $V^{-1}$, riportarlo sulla scala originaria.[33] Si evince, tuttavia, che tale metodo risulta particolarmente complesso laddove non sia nota la stima della deviazione standard e richiede un'ulteriore analisi, ovvero la valutazione preliminare del grado di correlazione tra tale statistica e lo stimatore del parametro.

Il metodo BCa, invece, è l'evoluzione del metodo Bootstrap *Bias Corrected* (BC), che considera nella valutazione degli intervalli di confidenza la presenza di un bias. In particolare, il metodo BC parte dall'ipotesi che esista una funzione monotona crescente g(.) tale che $\hat{\phi}=g(\hat{\theta})$ e $\phi=g(\theta)$, in virtù della quale sia verificata la seguente relazione[33]:

$$\hat{\phi}-\phi \sim N(-z_0\,\sigma,\ \sigma^2)$$

per cui $$\hat{\phi}^*-\hat{\phi} \sim \hat{\phi}-\phi \sim N(-z_0\,\sigma,\ \sigma^2)$$

Se non ci fosse il bias, definita come $F_{\hat{\theta}}(s)=\Pr\{\hat{\theta}\le s\}$, si avrebbe, in virtù della monotonicità della funzione g(.), che:

$$\Pr\{\hat{\theta}^*\le \hat{\theta}^*[1-\alpha]\}=\Pr\{\hat{\phi}^*\le \hat{\phi}^*[1-\alpha]\}=1-\alpha$$

Dato che $\Pr\{(\hat{\phi}^*-\hat{\phi})/\ \sigma\le z^{1-\alpha}\}= 1-\alpha=\Phi(z^{1-\alpha})$, si ottiene

$$\hat{\phi}^*[1-\alpha]= \hat{\phi}+ z^{1-\alpha},$$

da cui $$\hat{\theta}^*[1-\alpha]=g^{-1}(g(\hat{\theta})+z^{1-\alpha}).$$

Tale metodo è definito come *Metodo del percentile*.

Il metodo BC, invece, considera anche la asimmetria della distribuzione di partenza, attraverso l'introduzione del parametro $z_0$ di correzione del bias. In particolare, si suppone che:

$$\hat{\phi}-\phi \sim N(-z_0\sigma,\ \sigma^2)$$

per cui $$\hat{\phi}^*-\hat{\phi} \sim \hat{\phi}-\phi \sim N(-z_0\,\sigma,\ \sigma^2).$$

A partire da tale relazione, si ottiene, reiterando le valutazioni effettuate perl metodo del Percentile, che :

$$\hat{\theta}^*[1-\alpha]=g^{-1}(g(\hat{\theta})+z^{1-\alpha}+z_0).$$

Si evince che questa procedura, prescindendo dall'ipotesi di simmetria della distribuzione del parametro, presenta un errore di copertura inferiore rispetto al metodo del percentile. Tuttavia, anche in questo caso l'ipotesi di fondo è che la varianza sia costante. Al contrario, in genere si verifica che la varianza sia funzione del parametro da stimare. Per questo motivo, è stato introdotto il metodo BCa che tiene conto di un altro termine *a*, definito come coefficiente di accelerazione. In particolare, in questo caso si ipotizza che :

$$\hat{\phi}-\phi \sim N(-z_0\ \sigma_\phi,\ \sigma_\phi{}^2)$$

per cui $\hat{\phi}^*-\hat{\phi} \sim \hat{\phi}-\phi \sim N(-z_0\ \sigma_\phi,\ \sigma_\phi{}^2)$ con

$\sigma_\phi=1+a\phi$

Risulta, quindi, necessario considerare la scala stabilizzata della varianza, che può essere definita come nel caso del Bootstrap–t. In particolare, la scala da considerare può essere ottenuta a partire da V(x)=(1/a)log(1+ax). In altri termini, si considera ξ=h($\phi$) al fine di analizzare il problema trasformato, che risulta essere $\hat{\xi} \sim$ ξ +(1/a)log(1+a(Z-$z_0$)) con Z~ N(0,1)[BOOT8]. Dopo aver costruito l'intervallo esatto su questa scala, è possibile considerare le due trasformazioni inverse, ottenendo in tal modo il valore :

$$\hat{\theta}^*[1-\alpha]=g^{-1}(g(\hat{\theta})+[(z^{1-\alpha}+z_0)(1+ag(\hat{\theta}))/(1-a(z^{1-\alpha}-z_0)]).$$

Dato che

$$F_{\hat{\theta}^*}(s)= \Pr\{\hat{\theta}^* \le s\}=\Phi\{[g(s)-g(\hat{\theta})/(1+ag(\hat{\theta}))]+z_0\},$$

si definisce l'estremo dell'intervallo di confidenza, senza dover definire la funzione *g* come[55]:

$$\hat{\theta}^*[1-\alpha]= F_{\hat{\theta}^*}{}^{-1}(\Phi\ \{(z_0)+ ((z^{1-\alpha}+z_0)/(1-a(z^{1-\alpha}+z_0))\}).$$

Al contrario dell'intervallo standard, il Bca generalizza l'assunzione di normalità, almeno asintotica, considerando la presenza del bias, di un errore standard non costante e la trasformazione per la normalizzazione. In entrambi i casi, tali

assunzioni risultano non esatte; tuttavia il metodo tradizionale è sicuramente una peggiore approssimazione della realtà. Considerando i percentili alterati per correggerli rispetto al bias e all'accelerazione,inoltre, gli estremi dell'intervallo sono ottenuti attraverso l'inversa della funzione di distribuzione. Ne consegue che tali intervalli conservano il range del parametro. Per esempio, se il parametro varia tra 0 e 1, allora i limiti dell'intervallo rispetteranno tali vincoli. Infine, il BCa, a differenza del metodo standard e del Bootstrap-t, risulta essere invariante alle trasformazioni. In altri termini, se si considerano il parametro di interesse pari ad una funzione $\tau=m(\theta)$ e la sua stima $\hat{\tau}=m(\hat{\theta})$, allora gli estremi dell'intervallo possono essere calcolati a partire da $\hat{\theta}$ e $\hat{\theta}^*$ semplicemente considerando:

$$\hat{\tau}^*_{[1-\alpha]}=m(\hat{\theta}_{[1-\alpha]}^*)$$

Lo svantaggio di tale metodo è che l'errore di copertura cresce per α che tende a zero. Infatti, in tal caso si verifica che:

$$\Phi\{z_0+((z^{1-\alpha}+z_0)/(1-a(z^{1-\alpha}+z_0))\}\rightarrow \Phi\{z_0+1/a\}\neq 1$$

Il BCa può essere impiegato sia nel caso di Bootstrap parametrico che nonparametrico. In particolare, nel caso di Bootstrap parametrico la distribuzione di partenza è $F_{\hat{\theta}}=F(\mathbf{Y}, \hat{\theta})$ dove $\hat{\theta}$ è lo stimatore di massima verosimiglianza. Tale considerazione può essere ampliata al caso di modelli multiparametrici, considerando $\boldsymbol{\eta}$ il vettore dei parametri, $\theta=t(\boldsymbol{\eta})$ il parametro da stimare e $\hat{\eta}$ il vettore dei parametri stimati con il metodo della massima verosimiglianza. In questo caso, si effettua il ricampionamento a partire da $F_{\hat{\eta}}$.. Nel caso nonparametrico, invece, la distribuzione di partenza è quella empirica. In particolare, per il metodo non parametrico, effettuare il ricampionamento a partire dalla distribuzione empirica è equivalente all'estrazione con rimessa a partire dal campione originario.

L'algoritmo impiegato per la costruzione di un intervallo di confidenza pari ad (1-2 α) con il metodo BCa consiste, sia nel caso parametrico che non-parametrico, nella stima dei fattori di correzione $z_0$ e $a$ e nella conseguente determinazione dell'intervallo come:

$$(\hat{\theta}^*[\alpha],\ \hat{\theta}^*[1-\alpha])$$

con $\hat{\theta}^*[1-\alpha]= F_{\hat{\theta}^*}^{-1}(\Phi\ \{(z_0)+\ ((z^{1-\ \alpha}+z_0)/(1-a(z^{1-\ \alpha}+z_0))\}$

e $\hat{\theta}^*[\alpha]= F_{\hat{\theta}^*}^{-1}(\Phi\ \{(z_0)+\ ((z^{\ \alpha}+z_0)/(1-a(z^{\ \alpha}+z_0))\}).$

Per quanto concerne la stima di $z_0$, si verifica che, essendo $\Pr\{\hat{\phi} < \phi\}= \Phi(z_0)= F_{\hat{\theta}^*}(\hat{\theta})$, è possibile considerare:

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\sum_{b=1}^{B} I(\hat{\theta}_b^* < \hat{\theta})}{B}\right).$$

Per il parametro $a$, invece, è necessario distinguere il caso parametrico da quello non parametrico. Nel Bootstrap parametrico il coefficiente $a$ può essere definito come la skewness della funzione $\dot{l}_\theta(\hat{\theta}) = \frac{\partial \log(f_\theta(\hat{\theta}))}{\partial\theta}$ con $f_\theta(\hat{\theta})$ la funzione densità di distribuzione. La stima di $a$ può essere definita valutando la funzione $\dot{l}_\theta(\hat{\theta})$ con $\hat{\theta}=\theta$ come [54] :

$$\hat{a} = \frac{SKEW(\dot{l}_{\theta=\hat{\theta}}(\hat{\theta}))}{6}$$

Nel caso di Bootstrap parametrico per modelli multiparametrici, inoltre, supposto che il vettore $\boldsymbol{\eta}$ sia k-dimensionale, si considera la matrice $I_{\boldsymbol{\eta}}$ di dimensione $k$ x $k$, calcolata per $\boldsymbol{\eta}=\hat{\eta}$,con singolo elemento pari a:

$$a_{ij}= \frac{\partial^2 \log(f_\eta)}{\partial\eta_i\partial\eta_j}.$$

Inoltre, definito come $\nabla$ il gradiente di $\theta=t(\boldsymbol{\eta})$ valutato in $\hat{\eta}$, la minore direzione favorevole attraverso $\hat{\eta}$ risulta essere:

$$\hat{\mu}=(\ I_{\boldsymbol{\eta}})^{-1^*}\nabla$$

Ne consegue che la minore famiglia favorevole $\hat{F}$, ovvero quella per la quale l'informazione per θ(λ)=t($\hat{\eta}$+λ$\hat{\mu}$) a λ=0 è la stessa che per θ=t(**η**) nel problema k-dimensionale originario,. è la subfamiglia unidimensionale di $F_{\hat{\eta}}$ passante attraverso $\hat{\eta}$ nella direzione $\hat{\mu}$, definita come:

$$\hat{F}\colon\ f_{\hat{\eta}+\lambda\hat{\mu}} \qquad \text{con } \lambda \text{ parametro}$$

della famiglia.

In questo modo, il problema è ricondotto a quello con un unico parametro. In altri termini, il ricampionamento è effettuato a partire da $F_{\hat{\eta}}$ e la stima di $z_0$ non varia. Per quanto concerne, invece, il coefficiente di accelerazione, può essere definito come:

$$a = \frac{SKEW_{\lambda=0}\, d\log(f_{\hat{\eta}+\lambda\hat{\mu}})}{6d\lambda}$$

A questo punto, a partire dall'intervallo ($\lambda_{inf}$, $\lambda_{sup}$) si ottiene l'intervallo di confidenza attraverso la relazione θ(λ)=t($\hat{\eta}$+λ$\hat{\mu}$)[BOOT8].

Infine, nel caso di Bootstrap nonparametrico è necessario considerare la funzione empirica di influenza della statistica $\hat{\theta}$=t($\hat{F}$). Tale funzione può essere definita come[54]:

$$U_i = \lim_{\varepsilon\to 0} \frac{t(1-\varepsilon)\hat{F} + \varepsilon\delta_i)}{\varepsilon} \text{ con i=1,2...,n}$$

In questo caso, $\delta_i$ è una massa puntuale su $y_i$, tale che (1-ε)$\hat{F}$+ε $\delta_i$ è una versione di $\hat{F}$ con un maggior peso su $y_i$ e minore sugli altri punti. La stima di *a* è, quindi:

$$\hat{a} = \frac{\sum_{i=1}^{n} U_i^3}{6(\sum_{i=1}^{n} U_i^2)^{3/2}}$$

In realtà, da un punto di vista applicativo, si considera $U_i$ come la funzione di influenza Jackknife definita come:

$$U_i=(n-1)(\ \hat{\theta}-\hat{\theta}_{(\sim i)})$$

con $\hat{\theta}_{(\sim i)}$ pari alla stima del parametro basata sul vettore $\mathbf{y}=\{y_1,y_2,\ldots,y_{i-1},y_{i+1},\ldots,y_n\}$. In altri termini, lo stimatore può essere espresso ricorrendo direttamente ai valori dei $\hat{\theta}$ come [74]:

$$\hat{a} = \frac{\sum_{i=1}^{n}(\hat{\theta} - \hat{\theta}_{\tilde{(i)}})^3}{6\left\{\sum_{i=1}^{n}(\hat{\theta} - \hat{\theta}_{\tilde{(i)}})^2\right\}^{3/2}}$$

La scelta del metodo BCa è relazionata ai maggiori vantaggi che questa procedura presenta rispetto al Bootstrap-t in termini di invarianza rispetto alle trasformazioni e indipendenza dalla conoscenza dello stimatore della deviazione standard. In particolare, il Bootstrap-t è definito come un *pivotal method* riportando l'analisi a statistiche simili a quelle impiegate per le funzioni ancillari. Tra le tecniche appartenenti alla famiglia dei *nonpivotal method*, si è deciso di impiegare il BCa, piuttosto che il metodo del Percentile o il BC, in relazione alla maggiore generalità fornita da tale metodo.

Risulta, infine, necessario tenere in considerazione che vi sono altre tecniche definite come *Test-inversion interval* che si basano sulla dualità tra gli intervalli di confidenza e i test di ipotesi. Tale tecniche non sono state analizzate poiché risultano impiegabili solo nel caso parametrico. Inoltre, queste metodologie sono utilizzate quasi esclusivamente in relazione ai problemi di regressione.

Un ulteriore aspetto da analizzare è la possibilità di migliorare l'accuratezza degli intervalli di confidenza, che nel caso del BCa significherebbe ottenere un'accuratezza del terzo ordine, attraverso l'impiego di una tecnica Bootstrap definita *calibrazione*. Considerato $\hat{\theta}^*[1-\alpha]$ il limite superiore dell'intervallo di confidenza, se il metodo di definizione di tale intervallo risultasse perfetto, si avrebbe una probabilità di copertura effettiva esattamente pari ad α, ovvero:

$$\beta(\alpha) \equiv \Pr\{\hat{\theta} < \hat{\theta}^*[1-\alpha]\} = \alpha$$

Poiché questa relazione in genere non è verificata, è possibile usare la curva di calibrazione β(α) al fine di migliorare l'intervallo di confidenza. Per esempio, per un intervallo con 1-2α=0,90, se β(0.03)=0.05 e β(0.98)=0.95, allora è possibile considerare l'intervallo ($\hat{\theta}^*[0.03]$, $\hat{\theta}^*[0.98]$) piuttosto che ($\hat{\theta}^*[0.05]$, $\hat{\theta}^*[0.95]$).

Poiché la curva di calibrazione risulta incognita, è possibile utilizzare la tecnica Bootstrap per stimare β(α) come:

$$\hat{\beta}(\alpha) = \Pr_{*}\left\{\hat{\theta} < \hat{\theta}^{*}[1-\alpha]\right\}$$

dove $\Pr_*$ indica il campionamento Bootstrap. In altri termini, per un dato campione Bootstrap si considera $\hat{\alpha}^*$ come il valore di α tale che $\hat{\theta}^*[\hat{\alpha}^*]=\hat{\theta}$. Essendo gli eventi $\{\hat{\alpha}^* < \alpha\}$ e $\{\hat{\theta} < \hat{\theta}^*[\hat{\alpha}^*]\}$ equivalenti, si ha che:

$$\hat{\beta}(\alpha) = \Pr_{*}\{\hat{\alpha}^* < \alpha\}$$

Al fine di calibrare l'intervallo, si generano *B* campioni Bootstrap e si calcola per ciascuno di questi il valore $\hat{\alpha}^*$. La curva di calibrazione stimata si ottiene, attraverso la stima della funzione di distribuzione di $\hat{\alpha}^*$, come segue:

$$\hat{\beta}(\alpha) = \#\{\hat{\alpha}^* < \alpha\} / B$$

Se $\hat{\beta}(\alpha)$ non è circa uniforme, ovvero non assume valore pari ad α, allora l'intervallo può essere migliorato impiegando la stima della calibrazione. Nel caso del Bca, la calibrazione porterebbe ad un errore di copertura pari ad $O(n^{3/2})$, ovvero ad un'accuratezza del terzo ordine. Tuttavia, la calibrazione in questo caso richiederebbe un incremento eccessivo del numero di rivalutazioni della statistica originaria. In genere, quindi, tale metodo è impiegato nel caso di tecniche *Approximate Bootstrap Confidence intervals* (ABC). Tale metodo è un'approssimazione analitica del BCa, in quanto non richiede la valutazione della funzione di distribuzione empirica ma di un ulteriore coefficiente, oltre ad $a$ e $z_0$, ovvero del parametro di nonlinearità $c_q$ .. Questo metodo non è stato precedentemente illustrato, poiché è impiegato soprattutto nell'ambito di Bootstrap parametrici, anche se sono state sviluppate estenzioni al caso non-parametrico, per problemi che richiederebbero un onere computazionale eccessivo nel caso di impiego del BCa caratterizzati da distribuzione degli output appartenenti alla famiglia esponenziale, quale ad esempio Normale, Poisson, Binomiale o Gamma[54].

In realtà, il metodo Bootstrap è ampiamente utilizzato anche in termini di definizione del parametro che meglio rappresenta gli output di simulazione. In questo caso, il metodo Bootstrap è applicato ad una statistica *T* che rappresenta la *goodness-of-fit* del modello selezionato, come ad esempio la statistica Anderson-

Darling[12]. A partire dalla distribuzione Bootstrap della statistica, ottenuta da *B* campioni Bootstrap si considera il p-value come:

$$p=(\#T^{*(i)}\geq T)/B$$

Se il valore di *p* risulta troppo ridotto il modello è rigettato. Naturalmente, tale procedura può essere estesa alla scelta del miglior modello, considerando per ciascuno il p-value e selezionando quello con il valore maggiore [43]. Infine, il Bootstrap può essere impiegato anche per dati dipendenti. Infatti, spesso risulta complesso o non conveniente ottenere più replicazioni indipendenti. Una possibile soluzione è l'impiego di tecniche tradizionali che ricorrono ad un'unica replicazione, come ad esempio il metodo Batch Means o il metodo delle Serie temporali. Tuttavia, il primo metodo risulta complesso in termini di definizione della dimensione dei Batch e il secondo richiede delle assunzioni sul modello di riferimento che spesso non sono verificate. Tra le tecniche Bootstrap sviluppate con l'impiego di una sola replicazione per dati dipendenti, si considerano il metodo del *moving block* che si basa sulla divisione in blocchi adiacenti di lunghezza fissata l'insieme dei dati. Pseudo-dati sono creati concatenando blocchi scelti attraverso il ricampionamento senza reimmissione. Il problema fondamentale di tale tecnica è l'assunzione che i dati siano *n-dipendenti*, ovvero solo gli ultimi *m* dati influenzano il dato corrente. Queste assunzioni non sono verificate in molti casi, quando la funzione di autocorrelazione decresce lentamente, come ad esempio nel caso di sistemi di code complesse. Un terzo metodo, il *Bootstrap Stazionario*, consiste nel ricampionamento dei dati attraverso blocchi concatenati il cui punto di inizio è determinato in maniera casuale e la cui lunghezza è geometricamente distribuita con una media predefinita *p*. Gli ultimi due metodo illustrati presentano entrambi il problema della definizione della dimensione dei blocchi. Un ultimo metodo non-parametrico per l'analisi di dati correlati è *Bootstrap binario* per dati binari, descritto da Kim, Haddock e Willemain. Questa tecnica consiste nel ricampionamento alternativamente da runs di zero e uno compresi inciscuna serie binaria. Questa tecnica di ripartizione dei dati non richiede complesse valutazioni in merito alla dimensione dei batch. Inoltre, attraverso l'analisio empirica di alcuni sistemi, quali ad esempio un sistema di code M/M/1 con carico elevato e uno D/M/10, si è dimostrato che tale tecnica fornisce risultati spesso migliori rispetto a quelli ottenuti con il Batch Means. A partire da tale metodo è stata sviluppata una sua

generalizzazione definita come il *Threshold Bootstrap*. Tale procedura consiste nella definizione del threshold value, come ad esempio la media, si divide la serie in runs che sono o superiori o inferiori al threshold e si crea una replicazione Bootstrp comncatenandi i runs scelti alternativamente dalle popolazioni di alti runs o dei runs bassi. Si procede, quindi, al ricampionamento casuale con reimmissione dei runs. I runs concatenati sono troncati quando la lunghezza totale eccede N. Si valuta la statistica desiderata, per esempio la media campionaria, e si reitera il procedimento *B* volte. Le statistiche ottenute sono, quindi, analizzate come se i dati fossero indipendenti[99].

## 5.2. Il problema del numero di ricampionamenti nel Bootstrap

La distribuzione campionaria di una statistica mostra la variazione della statistica dovuta alla selezione casuale di campioni dalla popolazione. Infatti, si possono ottenere determinazioni della statistica differenti al variare del campione esaminato. Nel caso dell'impiego della distribuzione Bootstrap come distribuzione campionaria, si introduce un'ulteriore fonte di variabilità dovuta al ricampionamento dal campione originario. Si consideri, ad esempio, la distribuzione della popolazione in figura. Tale distribuzione risulta chiaramente non Normale presentando due picchi.

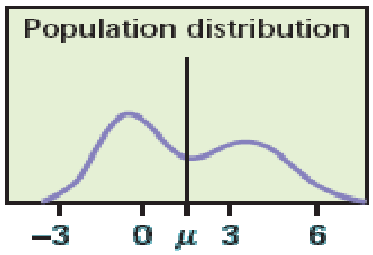



**Figura 5.4 Esempio di distribuzione di una popolazione**

Nell'ipotesi in cui il parametro da stimare sia la media della distribuzione, se si considera l'approccio tradizionale, è necessario estrarre dalla popolazione di partenza un numero sufficientemente elevato di campioni per valutare la distribuzione campionaria della statistica. Nella figura è rappresentata la distribuzione campionaria nel caso ideale in cui siano estratti dalla popolazione

tutti i possibili campioni. In virtù del teorema del limite centrale, tale distribuzione risulta palesemente Normale con media pari al parametro da stimare.

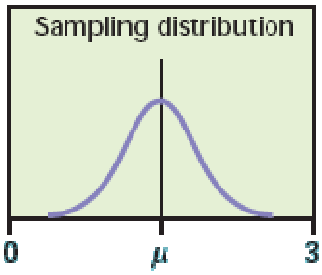



**Figura 5.5 Esempio di distribuzione campionaria**

Si considerino, cinque campioni estratti dalla popolazione originaria con dimensione campionaria pari a 50. Per ognuno di questi, è stata considerata la distribuzione Bootstrap sulla base di 1000 ricampionamenti . La figura mostra la distribuzione del campione originario e quella Bootstrap corrispondente.

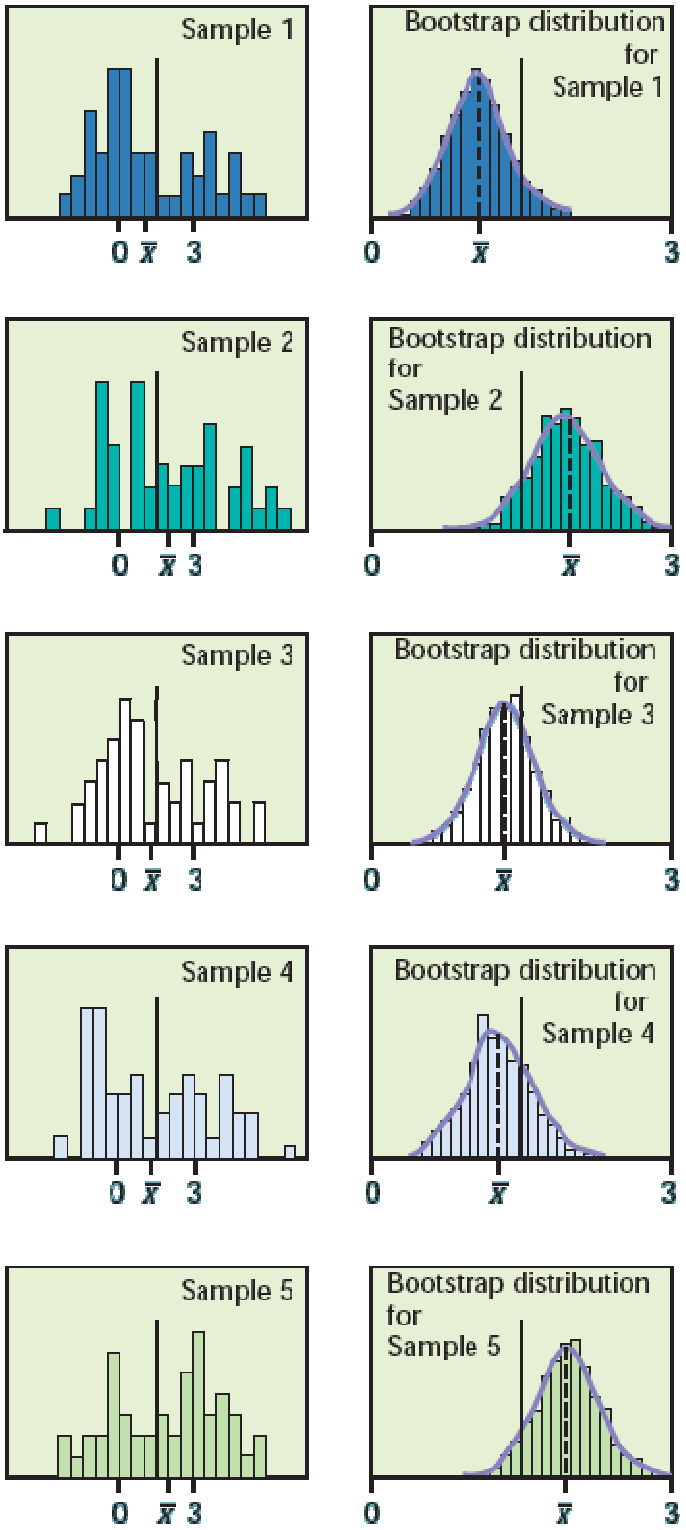



**Figura 5.6 Confronto tra cinque distribuzioni bootstrap**

Il confronto tra cinque distribuzioni bootstrap permette di analizzare l'effetto del campionamento casuale dalla popolazione di partenza, ovvero per i campioni

originari. Se ne deduce che ogni distribuzione Bootstrap è centrata in prossimità del valore della statistica per il campione originario, mentre la distribuzione campionaria è incentrata sul parametro incognito. Inoltre, la forma e l'ampiezza tra le diverse distribuzioni Bootstrap variano poco. Se ne deduce che queste caratteristiche, pur dipendendo dal campione originario, non variano tra i diversi campioni in maniera consistente. Quindi, la forma e l'ampiezza delle diverse distribuzioni Bootstrap è prossima a quella campionaria non ideale, presentando un bias simile a quello della distribuzione campionaria rispetto al parametro e la loro ridotta variabilità è dovuta al fatto che l'effetto del campionamento è contenuto per i campioni di partenza. Per valutare, invece, la variabilità connessa al ricampionamento, si considerano sei distribuzioni Bootstrap a partire dallo stesso campione originario. Le distribuzioni sono mostrate in Figura 5.7.

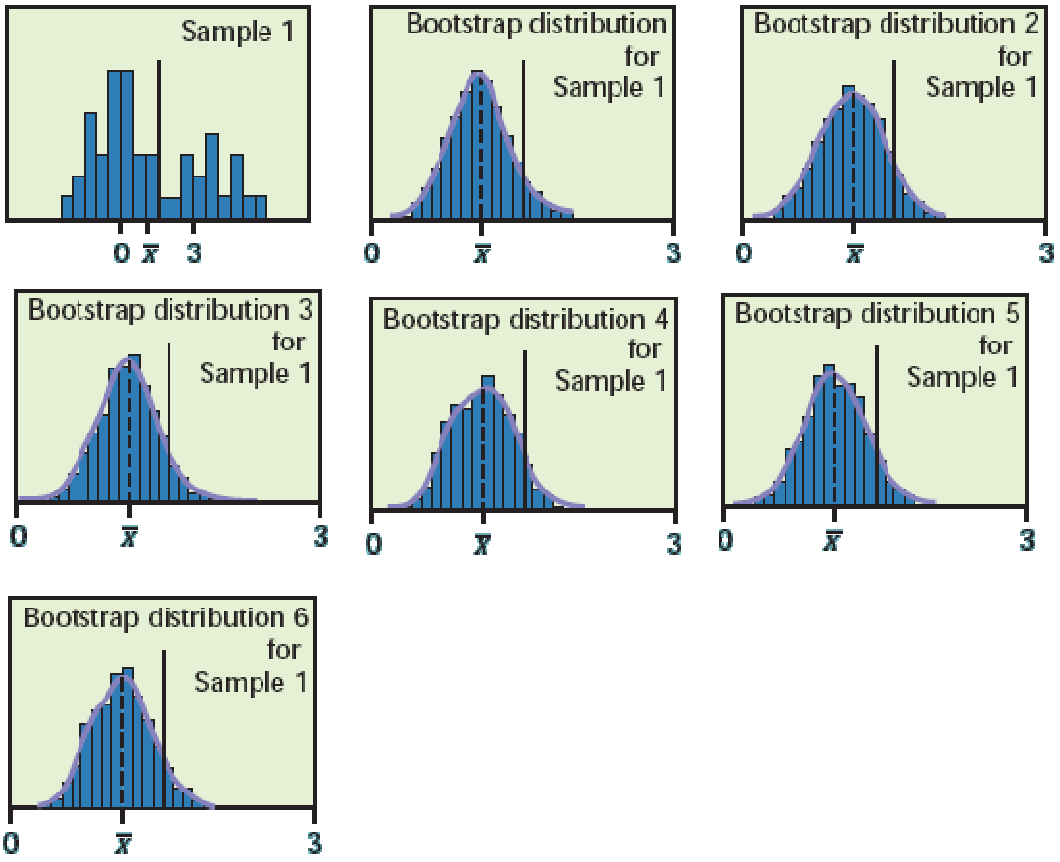



**Figura 5.7. Distribuzioni bootstrap ottenute a partire da uno stesso campione.**

Dalla Figura 5.7 si evince che le sei distribuzioni Bootstrap dallo stesso campione sono molto simili in termini di forma, ampiezza e centro. Di conseguenza, è possibile affermare che il ricampionamento aggiunge un errore modesto.
A questo punto, è possibile analizzare il diverso comportamento in relazione a due fattori, ovvero la dimensione del campione originario e il numero di campioni Bootstrap. Per valutare il primo elemento si è considerato una distribuzione della popolazione di partenza Normale, in modo da poter analizzare la distribuzione campionaria effettiva, che risulta comunque Normale anche per un numero ridotto di osservazioni. La dimensione dei cinque campioni originari è stata scelta pari a 9 e la variazione dovuta al campionamento dalla popolazione è stata analizzata confrontando le distribuzioni Bootstrap per i diversi campioni di partenza con un numero di campioni Bootstrap sempre pari a 1000.
Dalla figura si evince che le distribuzioni Bootstrap non sono più tutte simili in termini di forma e ampiezza alla distribuzione campionaria. Di conseguenza, nel caso di una dimensione ridotta del campione di partenza, l'errore dovuto al campionamento è elevato e, quindi, non è possibile analizzare la distribuzione Bootstrap per ottenere informazioni sulla distribuzione campionaria. Il confronto tra le diverse distribuzioni Bootstrap ottenute dallo stesso campione, invece, fornisce gli stessi risultati ottenuti nel caso precedente poiché il numero di campioni Bootstrap utilizzati per ciascuna distribuzione non è variato. Per quanto concerne la determinazione del numero di campioni Bootstrap, invece, è necessario definire la numerosità campionaria in funzione degli obiettivi dell'impiego della distribuzione Bootstrap. Ad esempio, in genere il numero di campioni Bootstrap per determinare l'errore standard è di gran lunga inferiore rispetto a quello minimo necessario per la valutazione degli intervalli di confidenza.
Efron e Tibshirani [61] hanno determinato il numero di campioni Bootstrap da impiegare per la valutazione dell'errore standard considerando il coefficiente di variazione delle stime. In altri termini, siccome la stima dell'errore standard effettivo realizzato sulla base della distribuzione campionaria $S\hat{E}(\hat{\theta})$ ha coefficiente di variazione non nullo, la stima $S\hat{E}^{*}(\hat{\theta})$ ottenuta dalla distribuzione Bootstrap avrà un coefficiente superiore dovuta all'ulteriore fonte di variabilità connessa al ricampionamento.

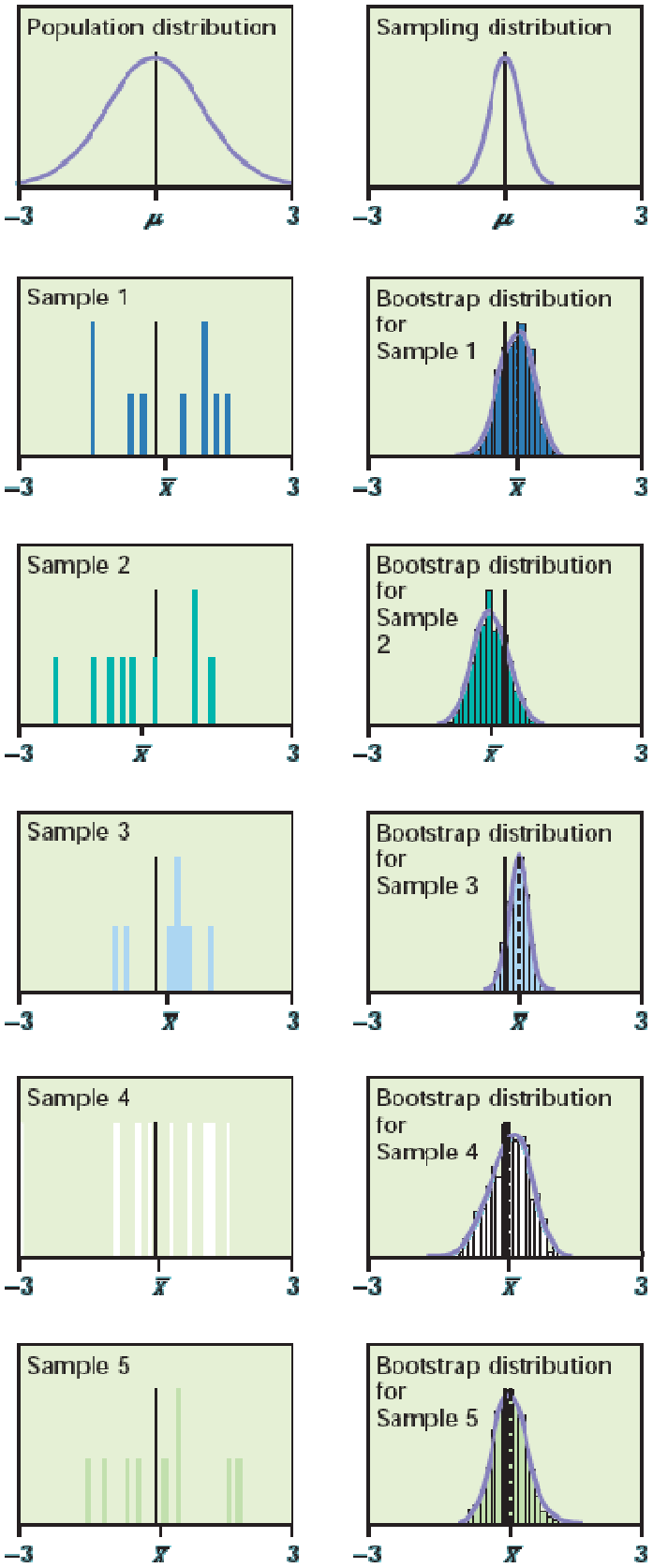



**Figura 5.8. Confronto tra distribuzioni bootstrap al variare del campione di partenza**

Il coefficiente di variazione, in particolare, è una misura relativa di dispersione. Tale coefficiente può essere calcolato considerando la seguente approssimazione:

$$CV(S\hat{E}*(\hat{\theta}) = \left\{ CV(S\hat{E}(\hat{\theta}))^2 + \frac{E(\hat{\delta}) + 2}{4B} \right\}^{1/2}$$

dove $\hat{\delta}$ è la curtosi della distribuzione Bootstrap. In particolare tale indice di forma, che rappresenta la concavità della distribuzione dei dati e può essere calcolato impiegando differenti tecniche statistiche, come l'indice di Fisher o quello di Pearson. Per quanto concerne il coefficiente di variazione, per una popolazione questo è pari a CV=(σ/μ)*100; nel caso di un campione, invece, si ha CV=(s/ $\bar{y}$)*100. Infine, se la dimensione campionaria è ridotta si considera il coefficiente corretto CV'=CV(1+(1/4n)) [157]. In genere, $CV(S\hat{E}(\hat{\theta}))$ varia tra 0,10 e 0,30, valori esterni a questo intervallo possono far sorgere il sospetto di un errore di rilevazione o di calcolo. Per esempio, se i dati hanno distribuzione Normale standard la dimensione del campione originario n è pari a 20 e lo stimatore è la media, allora il coefficiente è pari a 0,16. La tabella mostra diversi valori di $CV(S\hat{E}(\hat{\theta}))$ per diverse dimensioni $B$ e diverse $CV(S\hat{E}(\hat{\theta}))$ assunto $E(\hat{\delta})$=0. Per $CV(S\hat{E}(\hat{\theta}))$ > 0,10 il miglioramento è ridotto oltre B=100. Infatti, già con B=25 si ottengono risultati accettabili.

| | | B→ | | | | |
|---|---|---|---|---|---|---|
| | | 25 | 50 | 100 | 200 | ∞ |
| CV($S\hat{E}(\hat{\theta})$) | 0,25 | 0,29 | 0,27 | 0,26 | 0,25 | 0,25 |
| ↓ | 0,20 | 0,24 | 0,22 | 0,21 | 0,21 | 0,20 |
| | 0,15 | 0,21 | 0,18 | 0,17 | 0,16 | 0,15 |
| | 0,10 | 0,17 | 0,14 | 0,12 | 0,11 | 0,10 |
| | 0,05 | 0,15 | 0,11 | 0,09 | 0,07 | 0,05 |
| | 0 | 0,14 | 0,10 | 0,07 | 0,05 | 0 |

La situazione è differente nel caso degli intervalli di confidenza. Il calcolo effettuato da Efron mostra che B=1000 è approssimativamente un minimo per

calcolare gli intervalli BC e BCa. Per gli intervalli dati dal Bootstrap Percentile B=250 è approssimativamente sufficiente, in quanto non è necessario calcolare $z_0$. Per gli intervalli di confidenza, il numero di campioni necessario è maggiore poiché sono misure più complesse di valutazione dell'accuratezza statistica rispetto all'errore standard.
In figura troviamo l'andamento del limite inferiore di un intervallo di confidenza al valutato mediante BCa al variare del numero di ricampionamenti.

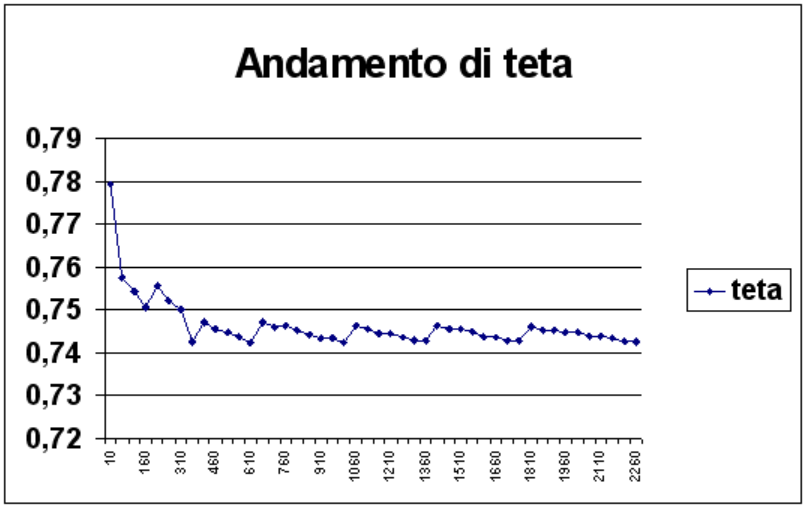



**Figura 5.9 Andamento del limite inferiore dell'intervallo di confidenza al variare del numero di ricampionamenti**

## 5.3. La validazione dei modelli di simulazione

I modelli di simulazione devono essere analizzati per determinarne la validità effettuandone la verifica e la validazione. La verifica del modello è spesso definita come l'accertamento che il modello computerizzato e la sua implementazione siano corretti [147]. La validazione di un modello di simulazione, invece, consiste nella valutazione del grado di rappresentatività di questo rispetto al sistema reale [91]. Nell'ambito della simulazione di sistemi stocastici, la validazione può essere anche intesa come il processo di costruzione di un livello accettabile di confidenza che consenta una corretta e valida inferenza da un processo simulato a un processo reale [170].
Ci sono quattro approcci basilari per valutare la validità di un modello. In ogni caso, la verifica e la validazione del modello di simulazione sono parte integrante

del processo di sviluppo del modello. Un primo approccio consiste nella verifica di validità effettuata direttamente dal team che ne realizza lo sviluppo sulla base di un'ampia gamma di test e valutazioni possibili. Un approccio più oggettivo, invece, è quello basato sul coinvolgimento dell'utilizzatore del modello nell'analisi di validità. Infine, è possibile considerare l'intervento di una terza parte nella validazione e verifica del modello, tale approccio è definito come *"indipendent verification and validation"*(IV&V). Questa tecnica è impiegata soprattutto nel caso di modelli di simulazione molto complessi e costosi su larga scala che spesso coinvolgono più teams per il loro sviluppo. La valutazione può essere effettuata sia in maniera concorrente allo sviluppo, in questo caso ciascuna fase dello sviluppo non procede fino a che non sono state effettuate la validazione e la verifica o dopo il completamento del modello. Naturalmente il primo approccio permette, laddove ci sia una sufficiente interazione tra il team e la terza parte, una riduzione notevole dei costi e dei tempi di sviluppo. Il quarto approccio proposto da numerosi autori tra i quali Balci, Gass et al., si basa su un modello a punteggio, nel quale si assegna dei pesi in maniera piuttosto soggettiva a diversi aspetti del processo di validazione e si combinano per ottenere dei punteggi di categoria e uno complessivo da confrontare con dei valori di riferimento. Tale approccio, in ogni caso, è poco usato nella pratica in quanto sia i pesi che i valori di riferimento sono definiti in maniera piuttosto soggettiva e spesso tale tecnica non evidenzia alcune limitazioni del progetto. Il processo di sviluppo del modello può essere rappresentato come in Figura 5.10 [147]

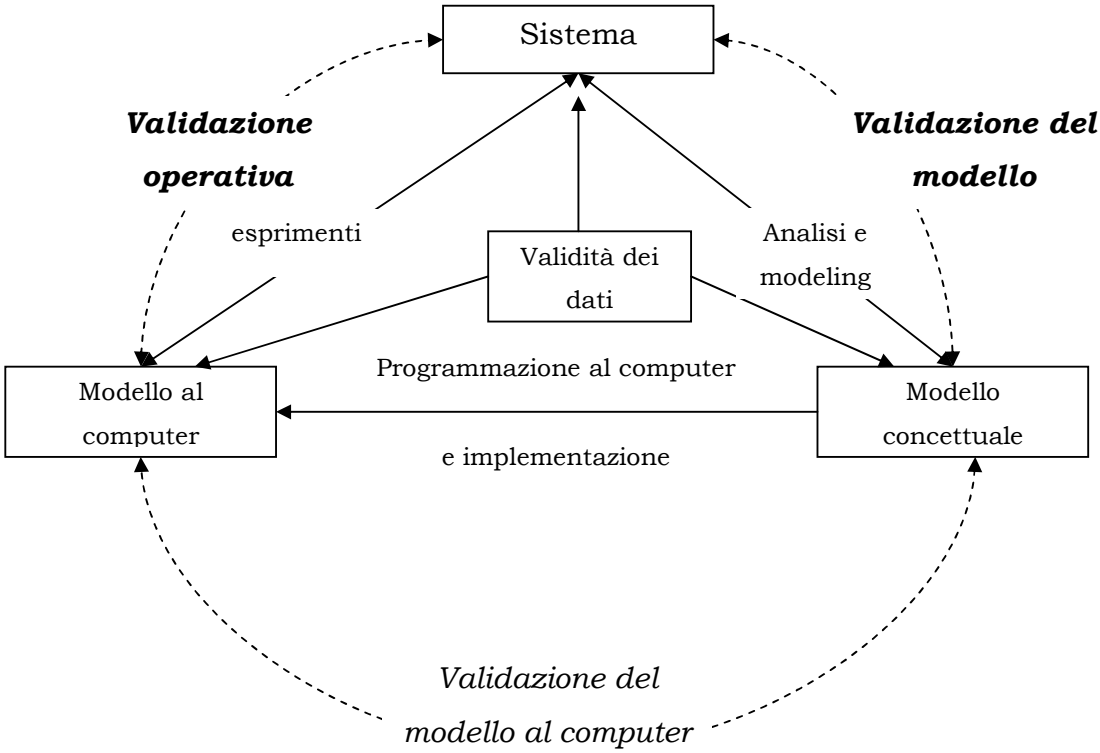



**Figura 5.10 Processo di sviluppo e validazione di un modello di simulazione**

Il sistema reale è rappresentato da un *modello concettuale*, realizzato durante la fase di analisi e creazione del modello, che ne costituisce una rappresentazione logica, verbale o matematica per un particolare studio. Tale modello è implementato al computer attraverso la fase di programmazione e implementazione per ottenere il *modello al computer*. Effettuando degli esperimenti su tale modello è possibile realizzare un'analisi inferenziale sul sistema di partenza. Dalla rappresentazione del processo di sviluppo si evince che la validazione del modello è realizzata in diverse fasi e spesso prima di ottenere un modello complessivamente valido sono sviluppate numerose versioni del modello. La prima validazione è effettuata in merito al modello concettuale per verificare se le assunzioni e le teorie alla base siano o meno corrette in relazione agli scopi dello studio, ovvero se queste risultano consistenti rispetto a quelle che descrivono le caratteristiche del sistema e il suo possibile comportamento e se la rappresentazione del sistema risulta "ragionevole" per gli scopi del progetto. La validazione del modello sviluppato al computer, invece, consiste nella verifica che la programmazione e implementazione del modello concettuale sia corretta. La

validazione successiva, ovvero quella operativa, permette di determinare se il comportamento degli output del sistema sia sufficientemente accurato rispetto al sistema reale in relazione all'applicabilità del modello simulato. Infine, l'analisi dei dati consiste nella verifica che i dati impiegati per la costruzione del modello, la sua valutazione e la realizzazione degli esperimenti siano adeguati e corretti.
In particolare, l'attenzione è focalizzata sulla fase di validazione operativa, in quanto si assumono come validi sia il modello concettuale che quello implementato al computer. Per effettuare la validazione operativa e, in generale, per gran parte delle validazioni necessarie nel processo di sviluppo del modello, possono essere impiegate diverse tecniche. In particolare, queste tecniche possono essere classificate come *soggettive* o *oggettive*. Esempi di tecniche oggettive sono quelle che impiegano test d'ipotesi o intervalli di confidenza. In particolare, nell'ambito della validazione operativa, le tecniche possono essere raggruppate come mostrato nella tabella. La Figura 5.11 mostra solo quali tecniche sono maggiormente diffuse nell'ambito della validazione dei modelli di simulazione, non tutte le possibili procedure realizzabili.

| | ***SISTEMA OSSERVABILE*** | ***SISTEMA NON OSSERVABILE*** |
|---|---|---|
| ***APPROCCIO SOGGETTIVO*** | ❖ Confronto impiegando strumenti grafici. | ❖ Valutazione del comportamento del modello. |
| ***APPROCCIO OGGETTIVO*** | ❖ Confronto impiegando procedure e test statistici. | ❖ Confronto con altri modelli impiegando procedure e test statistici. |

**Figura 5.11 Tecniche più diffuse per la validazione di modelli di simulazione**

Sebbene non ci sia nessun algoritmo che permetta di determinare quale tecnica impiegare, è comunque possibile effettuare una distinzione sulla base dell'osservabilità o meno del sistema, ovvero se è possibile collezionare dati sul comportamento del sistema reale o meno. In particolare, l'insieme delle tecniche

che sono basate sul *"confronto"* comprende sia le procedure che effettuano il paragone con gli output del sistema reale sia quelle che confrontano output di altri sistemi. Inoltre, le tecniche che impiegano la valutazione del comportamento del modello possono includere anche la variabilità dei parametri-analisi di sensibilità. In ogni caso, sia per il confronto che per la valutazione del modello, per avere un alto livello di confidenza, sarebbe necessario esplorare diversi insiemi di condizioni sperimentali, anche definiti come scenari, nel dominio di applicabilità del modello. Per questo motivo, nel caso di sistemi non osservabili il livello di confidenza risulta non molto elevato. In particolare, per ogni classe si considereranno brevemente le tecniche che possono essere impiegate, focalizzando successivamente l'attenzione solo sugli approcci di tipo statistico. Infatti, tali strumenti di analisi risultano maggiormente adeguati al caso in esame di processi con input e output aleatori e permettono anche in caso di analisi soggettiva di ottenere un grado di accuratezza e di generalità dell'analisi elevato.
La valutazione del comportamento del sistema è un insieme di tecniche che permettono un'analisi soggettiva di validità sia nel caso di sistemi osservabili che non osservabili. In particolare, nel primo caso la valutazione è quantitativa, nel secondo è di tipo qualitativo. In genere, tale approccio è maggiormente diffuso nel caso in cui non ci sia disponibilità di dati reali. In questo caso, è possibile considerare non la grandezza ma la direzione del comportamento degli output. Possibili approcci soggettivi non basati su tecniche statistiche sono l'*animation* e la *face validity*. La prima tecnica consiste nell'analisi del comportamento operativo attraverso una rappresentazione grafica dell'evoluzione del sistema nel tempo, per esempio i movimenti delle parti possono essere mostrati graficamente durante la simulazione. Nel secondo caso, invece, si raccolgono opinioni di esperti in merito al comportamento del sistema. In genere, tali analisi sono comunque poco diffuse nell'ambito della validazione operativa. Maggiormente utilizzate sono le tecniche statistiche, quali il *Design of experiment (DOE)* e la realizzazione di *Metamodelli.* Il DOE è impiegato nel caso in cui i dati reali risultano essere non disponibili o insufficienti. In questo caso, in genere, è nota l'influenza qualitativa degli effetti sul modello ma non quella quantitativa. In particolare è possibile realizzare almeno una *analisi di sensibilità* (o analisi what-if) [91]. Questa tecnica consiste nella valutazione sistematica delle risposte del sistema simulato a valori "estremi " degli input del modello o a drastici cambiamenti nella struttura del

modello. Per esempio, si considera il comportamento del modello di simulazione per un processo produttivo in relazione a sostanziali variazioni del tasso di arrivo dei jobs o a cambiamenti delle regole di priorità impiegate. Queste tecniche sono particolarmente adeguate, ad esempio, per realizzare i *degenerative tests* e gli *Estreme conditions test*. Il primo insieme di tests valuta la degenerazione del modello sotto adeguati input, per esempio si analizza se il numero di elementi in coda di un sistema di code continua ad aumentare se il tasso di arrivo è maggiore di quello di servizio. Il secondo metodo, invece, valuta se la struttura del modello e gli output possano essere plausibili per ogni condizione estrema, per esempio valuta se con un livello di scorte nulle l'output di produzione tende ad annullarsi. Nell'ambito del DOE, si definisce come fattore un parametro, una variabile in input o un modulo del modello di simulazione. L'analisi di sensibilità permette di effettuare la validazione del modello in quanto mostra se i fattori hanno effetti in accordo con il comportamento qualitativo atteso del sistema. Allo stesso tempo, tale analisi evidenzia quali fattori sono rilevanti. In genere, questo tipo di informazioni dovrebbero essere analizzate anche nel caso in cui siano disponibili dati relativi al sistema reale, in quanto se i fattori significativi sono controllabili l'analisi di sensibilità permette di valutarne la migliore configurazione per ottimizzare le prestazioni del sistema reale. L'analisi di sensibilità richiede la generazione di un insieme di run di simulazione ciascuno realizzato sotto fattori costanti. L'analisi può essere realizzata modificando un fattore per volta, ma tale tecnica non permette di considerare l'interazione tra i fattori, ovvero gli effetti dovuti a cambiamenti simultanei. Invece, il DOE con risoluzione 4 o 5 permette di considerare anche la stima dell'interazione di due fattori. Il problema centrale del DOE è la selezione delle combinazioni di livelli dei fattori da osservare tra tutte le possibili. Una possibilità è quella di realizzare piani fattoriali ridotti o di effettuare l'analisi del piano centrale. Una volta effettuata la selezione si realizza la simulazione e si analizzano i dati utilizzando tecniche quali l'*ANOVA* o l'*analisi di regressione*. Un esempio di modelli di regressione sono i metamodelli. Essi sono il modello del comportamento I/O del modello di simulazione. In genere, tali metamodelli usano una delle seguenti approssimazioni polinomiali:

- un'approssimazione del primo ordine polinomiale caratterizzato da una media complessiva $\beta_0$ e $k$ effetti principali $\beta_j$ con j=1,2,..,k.

- un'approssimazione del primo ordine polinomiale aumentata dall'interazione tra coppie di fattori $\beta_{j;j'}$ con j=1,2,..,k e j'=j+1,…,k;
- un'approssimazione del secondo ordine polinomiale che aggiunge gli effetti quadratici $\beta_{j;j}$ al caso precedente.

Naturalmente, la prima approssimazione trascura le interazioni tra fattorie ha effetti marginali costanti. Sarebbe possibile estendere l'approssimazione del secondo ordine ad una del terzo ordine, ma sarebbe di difficile interpretazione e richiederebbe un numero di runs molto elevato per stimare i molti parametri β. Quindi, l'approssimazione di secondo ordine risulta essere un buon compromesso tra accuratezza e onere computazionale. Per la scelta del grado di approssimazione e per la validazione del metamodello è possibile impiegare statistiche quali il coefficiente multiplo di correlazione $R^2$o procedure più accurate quali il DOE sequenziale combinato con la cross-validazione e il test F di Rao. In ogni caso, in quest'ambito sono stati sviluppati numerosi casi studio, tra i quali quelli che evidenziano il ruolo del DOE e dell'analisi di regressione nell'ambito della validazione in campo ecologico sviluppati da Kleijnen e da Kleijnen, Van Ham e Rotmans, o quello della simulazione sonar di Kleijnen.

Nel caso in cui non siano disponibili dati reali in input, inoltre, è possibile realizzare il confronto degli output del modello simulato con quelli di altri sistemi, ovvero con modelli di simulazione simili già validati o con modelli analitici. Possibili tecniche sono la *validità degli eventi*, che consiste nel confronto del numero di occorrenze di un determinato evento per il sistema simulato con quello del sistema reale; la *validazione di dati storici*, che richiede il confronto dei dati reali non utilizzati per la generazione del modello con quelli simulati e i *grafici operativi*, in questo caso il comportamento dinamico di alcuni indicatori di performance sono rappresentati graficamente per valutarne la correttezza. Le tecniche di confronto possono essere classificate in due categorie:

- Confronto grafico dei dati;
- Analisi statistica dei dati.

In particolare, l'ultima classe permette di effettuare valutazioni oggettive. Per questo motivo, in genere questa tecnica è impiegata. Tuttavia, in alcuni casi tale approccio non può essere impiegato, nel caso ad esempio in cui le assunzioni richieste dalle analisi statistiche non possono essere soddisfatte o i dati reali a

disposizione non sono sufficienti per realizzare analisi significative. In questi casi, si possono utilizzare i metodi grafici. In questo caso, il comportamento del modello simulato e del sistema è rappresentato graficamente per diversi insiemi di cindizioni sperimentali per determinare se il comportamento simulato è sufficientemente accurato per gli scopi per i quali è stato realizzato. È possibili impiegare tre tipi di grafici: istogrammi, box plots e grafici comportamentali utilizzando lo scatter plot. I grafici possono essere realizzati impiegando diverse misure, come ad esempio la media, la varianza, il massimo, la distribuzione e le serie di tempo delle variabili. Inoltre, è possibile considerare la relazione tra due misure di una variabile o tra diverse misure di due variabili. La scelta delle misure e delle relazioni da analizzare è spesso conseguente agli scopi della simulazione. I grafici in ogni caso non richiedono l'ipotesi di indipendenza delle variabili e non si richiede una particolare distribuzione, come ad esempio la normalità dei dati. Per quanto concerne le tecniche statistiche impiegate, queste procedure saranno analizzate soltanto in termini di confronto con il sistema reale. Infatti, nel caso in cui i dati reali non siano disponibili, in genere si preferisce impiegare tecniche di valutazione del comportamento del modello come il DOE piuttosto che confrontarlo con altri sistemi validi. In merito al confronto tra dati reali e simulati, supposti entrambi generati sotto scenari simili, si verifica che i dati reali possono essere sia dati in input che in output. A seconda della disponibilità di tali informazioni si impiegheranno diverse tipologie di tecniche di validazione. Laddove siano disponibili i dati reali in output, ad esempio quando il sistema è monitorato, ma gli scenari risultino non misurabili, allora è possibile effettuare il confronto tra dati in output reali e simulati. Si considerano quali output del sistema reale il vettore $W_{i;t}$ e quelli del sistema simulato $V_{j;t}$ con $i=1,2,\ldots,n$, $j=1,2,\ldots.m$ e $t=1,2,\ldots..,k$ e si faccia l'ipotesi di indipendenza stocastica e uguale distribuzione per gli elementi appartenenti a ciascun insieme di output. Per valutare le performance del sistema si considera un unico parametro , indicato come X e Y rispettivamente per il sistema reale e quello simulato. Il modello di simulazione ideale dovrebbe avere una funzione di distribuzione $F_y$ identica a quella del sistema reale $F_x$. Tuttavia, in genere è sufficiente che le medie siano uguali, ovvero $E(X)=E(Y)=\mu_x=\mu_y$. Se si considera il parametro $\mu_d=\mu_x-\mu_y$ e i classici stimatori $\bar{x}$, $\bar{y}$, $S^2_x$ e $S^2_y$ rispettivamente della media e della

varianza di x e y, allora è possibile considerare una statistica T-student con *n+m-2* gradi di libertà:

$$t_{n+m-2} = \frac{(\bar{x} - \bar{y}) - \mu_d [(n-m-2)nm]^{1/2}}{[(n-1)S_x^2 + (m-1)S_y^2]^{1/2}(n+m)^{1/2}}$$

Di conseguenza è possibile realizzare un test d'ipotesi, considerando come ipotesi nulla $H_0$: $\mu_d = 0$. La potenza del test aumenta al crescere di $|\mu_d|$, differenze maggiori sono più facili da rilevare, all'aumentare di *n* o *m* e al diminuire di $\sigma_x$ e $\sigma_y$, questo implica minor rumore. Il problema fondamentale di questo tipo di test è l'assunzione di normalità di X e Y. Un'alternativa a questo tipo di test è la statistica Student modificata di Johnson che include uno stimatore per la skewness delle distribuzioni in output. Ulteriori possibilità sono costituite dai tests distribution-free, come il rank test, l'impiego del Jackknife , anche se tali alternative sono raramente applicate nella pratica. Una valida alternativa è l'impiego del metodo Bootstrap.

Infine, nel caso in cui sono disponibili sia i dati in input che quelli in output relativi al sistema reale, è possibile effettuare la *trace-driven* simulation. Poichè il sistema in analisi presenta dati in input e output noti, è su questa tecnica che si focalizzerà maggiormente l'attenzione. La scelta dei metodi statistici che possono essere impiegati in funzione della disponibilità dei dati è riassunta nella Figura 5.12[91].

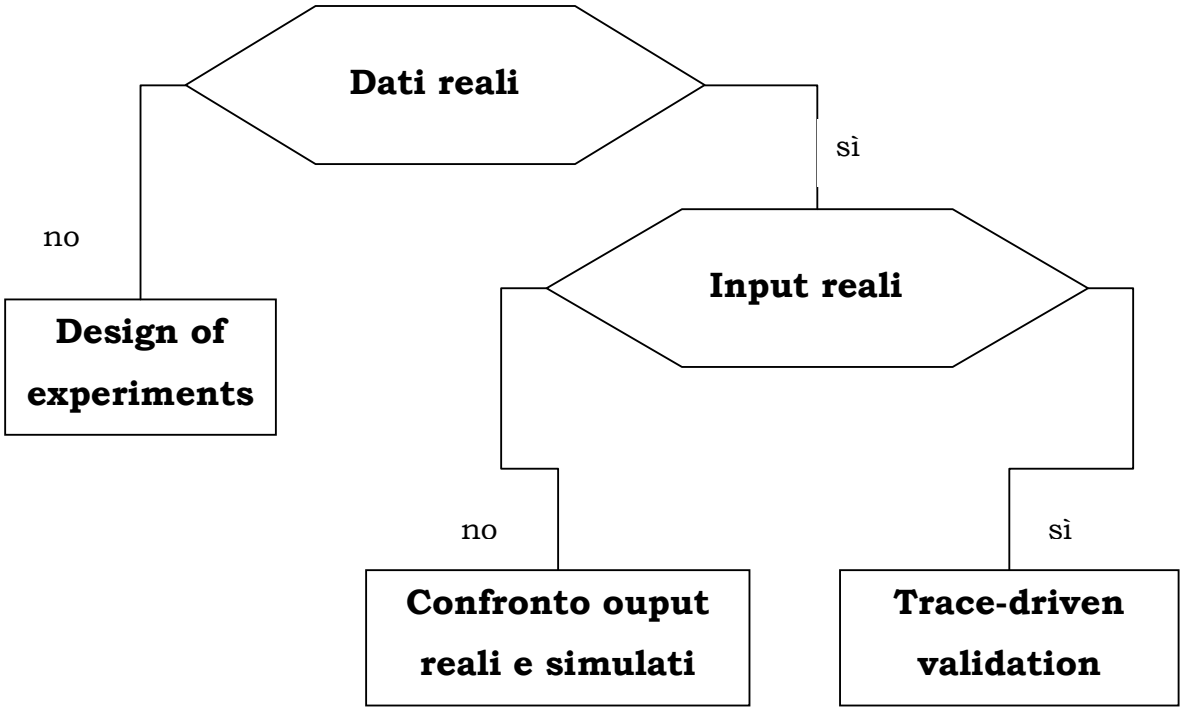



**Figura 5.12. Schema dei metodi statistici da utilizzare in base ai dati disponibili**

Nel caso della trace-driven simulation il sistema simulato è soggetto sia ad input reali, indicati come **A**, che risultano anche in ingresso al sistema reale, sia ad uno o più input **R** generati utilizzando i flussi di numeri pseudo-casuali. Un esempio della prima tipologia di dati in ingresso può essere la sequenza storica dei tempi di arrivo. Mentre, nel secondo caso è possibile considerare i tempi di servizio. Si considerano quali output del sistema reale il vettore $W_{i;t}$ e quelli del sistema simulato $V_{i;t}$ con $i=1,2,\ldots,n$ e $t=1,2,\ldots\ldots,k$. Per esempio, il tempo di completamento di ciascun job $t$ nel giorno $i$. Per valutare le performance del sistema si considera un unico parametro , indicato come X e Y rispettivamente per il sistema reale e quello simulato. Per effettuare la validazione, i due indici sono confrontati per gli stessi scenari caratterizzati dalla traccia **A**[96]. Le ipotesi di base per la validazione del modello è che sia l'insieme degli output reali X che quello delle risposte simulate Y siano i.i.d.. In particolare, questa ipotesi non risulta particolarmente restrittiva. Infatti, perché le variabili siano i.i.d. nel caso del sistema simulato è sufficiente che ogni subrun inizi nello stato vuoto e sia arrestato al verificarsi sempre dello stesso evento finale, trattandosi di una terminating simulation, come ad esempio dopo un numero fissato di job k. Il

problema della validazione del modello di simulazione è stato ampiamente trattato nel caso in cui si assuma un'ulteriore ipotesi di base, ovvero quella di distribuzione normale bivariata per le coppie $(X_i,Y_i)$. In questo caso, si considerano due possibili procedure. La prima tecnica si basa sull'assunzione che nel caso in cui il sistema simulato sia perfettamente identico a quello reale, allora il vettore $\mathbf{Y}=\{Y_1,Y_2,\ldots,Y_n\}$ dovrebbe coincidere con $\mathbf{X}=\{X_1,X_2,\ldots,X_n\}$, il che implica un coefficiente di regressione $\rho_{xy}=1$. Questa relazione può essere tradotta in termini di regressione lineare con l'ipotesi che il modello $y=\beta_0+\beta_1 x$ abbia coefficienti $\beta_0=0$ e $\beta_1=1$. Tuttavia, Kleijnen ([91] e [94]) ha dimostrato che tale procedura risulta molto spesso eccessivamente restrittiva. Infatti, il modello simulato può essere considerato valido nel caso in cui la media $\mu_y$ coincida con la media $\mu_x$ e che le varianze $\sigma^2_y$ e $\sigma^2_x$ siano uguali, anche se $\rho_{xy}$ è minore di uno e positivo. In questo caso, sulla base dell'ipotesi di distribuzione Normale Bivariata è possibile considerare valido il modello se $\beta_1=\rho_{xy}>0$ e $\beta_0=\mu(1-\rho_{xy})$ con $\mu=\mu_x=\mu_y$. Se e solo se $\rho_{xy}=1$ si ha che $\beta_0=0$ e $\beta_1=1$. Quindi, la tecnica precedente porta a rigettare anche modelli validi nella realtà. Per questo motivo, Kleijnen et al. introducono una tecnica innovativa che, supposta una correlazione positiva tra le X e le Y, valuta l'uguaglianza di media e varianza analizzando due statistiche ulteriori definite come $D_i=X_i-Y_i$ e $Q_i=X_i+Y_i$ . Si dimostra che le varianze sono uguali nel caso in cui le differenze D e le somme Q risultino incorrelate e che l'uguaglianza delle medie implica che il valore atteso di D sia nullo. Di conseguenza, la validazione è effettuata considerando la regressione di D su Q:

$$E(D/Q=q)=\gamma_0+\gamma_1 q$$

Sotto l'ipotesi di Normalità, l'uguaglianza delle varianze e delle medie si traducono rispettivamente in $\gamma_0=0$ e $\gamma_1=0$. In breve, si considera un'ipotesi nella $H_0$: $\gamma_0=0$ e $\gamma_1=0$. Per testare tale ipotesi, si considera la Somma degli Errori Quadratici (SSE) sotto l'ipotesi $H_0$ e quella alternativa $H_1$. Il SSE può essere calcolato nel caso di non validità di ipotesi nulla come:

$$SSE_{full}=\sum_{i=1}^{n}(D_i-\hat{D}_i)^2$$

con $\hat{D}_i=C_0+C_1Q_i$, dove $C_0$ e $C_1$ sono gli stimatori Ordinary Least Square di $\gamma_0$ e $\gamma_1$.

Nel caso di validità dell'ipotesi nulla, si ha che $\hat{D}_i=0$, da cui

$$SSE_{reduced} = \sum_{i=1}^{n} D_i^2$$

Queste due SSE forniscono una statistica Z di Fisher con due ( numero dei parametri di regressione per la D) e n-2( n osservazioni e due parametri stimati $C_0$ e $C_1$) gradi di libertà:

$F_{2,n-2}=[(n-2)/2][SSE_{reduced}-SSE_{full}]/SSE_{full}$. Se F è significativamente alta, allora si rigetta l'ipotesi nulla, ovvero si considera il modello non valido. Tale valutazione è stata, inoltre, testata su un sistema di code M/M/1 da Kleijnen et el.[94]. Il problema fondamentale di tale analisi consiste nell'ipotesi di Normalità dei dati. Tale ipotesi non è verificata soprattutto nel caso di subrun ridotti, ad esempio per k=10. Per questo motivo, è stata sviluppata da Kleijnen, Cheng e Bettonvil un'ulteriore tecnica che si avvale del metodo Bootstrap[96]. In alternativa al Bootstrap, sarebbe possibile sviluppare un numero sufficientemente alto di replicazioni al fine di considerare il teorema del limite centrale. Tuttavia, quest'ultima possibilità non è stata considerata poiché comporterebbe un incremento notevole dei tempi e dei costi di simulazione. Oltre alla indipendenza e all'identica distribuzione delle risposte sia del sistema simulato che di quello reale, si considera noto il numero di replicazioni s e inferiore rispetto al numero di campioni Bootstrap *b*. Kleijnen, Cheng e Bettonvil hanno considerato sei statistiche per la validazione del modello. La prima statistica $T_1$ corrisponde alla statistica valutata da Kleijnen nel caso di distribuzione Normale ed è impiegata per verificare l'uguaglianza delle varianze. Kleijnen, Cheng e Bettonvil hanno, inoltre, considerato le statistiche $T_2= \Sigma D_i /n$, definita come *deviazione media*, $T_3= \Sigma (Y_i /X_i)/n$ definita come l'*errore relativo medio*. Lo svantaggio della deviazione media è che errori positivi e negativi del modello possono compensarsi. Per ciascuna di tali statistiche è possibile considerare un valore di target sotto l'ipotesi nulla $H_0$. In particolare, il valore di riferimento per $T_2$ è zero e per $T_3$ il valore target è uno, nell' ipotesi che le $X_i$ non siano nulle. Si valutano, inoltre, anche le statistiche $T_4=\Sigma|D_i|/n$, definita come *errore di predizione medio assoluto*, $T_5=\Sigma D^2_i/n$, ovvero la *deviazione media quadratica (MSE)*, $T_6=\int_{-\infty}^{\infty}\left|\hat{F}_x(z)-\hat{G}_y(z)\right|dz$, che confronta le due distribuzioni empiriche dei dati reali e quelli simulati rispettivamente. Le ultime tre statistiche considerate sono valutate da Kleijnen, Cheng e Bettonvil sulla base di una distribuzione nota,

seppur non normale, degli output; nel caso in esame tali statistiche non saranno considerate. In ogni caso, tale limitazione risulta accettabile, in quanto la valutazione dell'uguaglianza delle medie è effettuata impiegando le statistiche $T_2$ e $T_3$.

La simulazione è effettuata in questo caso considerando *n* subruns di lunghezza fissata non sovrapposti, analizzati senza eliminare il transitorio e ciascuno soggetto ad un ingresso $\mathbf{A_i}$ con *i=1,2….,n*. In particolare, sia Kleijnen, Cheng e Bettonvil [96] che Kleijnen et al. [94], il secondo nel caso di distribuzione Normale, considerano due possibili valori di *n*, ovvero 10 e 25. Il numero di replicazioni *s* è variabile. In particolare, si effettua l'analisi considerando tre casi:

- *s*=1;
- *s*=2;
- *s*>2.

Il piano della simulazione è schematizzato nella Figura 5.13.

| | Subrun 1 | | Subrun i | | Subrun n |
|---|---|---|---|---|---|
| Traccia: | $A_1$ | … | $A_i$ | … | $A_n$ |
| Performance reali: | $X_1$ | … | $X_i$ | … | $X_n$ |
| Replicazione 1 | $Y_1^{(1)}$ | … | $Y_i^{(1)}$ | … | $Y_n^{(1)}$ |
| ….. | | | | | |
| Replicazione r | $Y_1^{(r)}$ | … | $Y_i^{(r)}$ | … | $Y_n^{(r)}$ |
| …. | | | | | |
| Replicazione s | $Y_1^{(s)}$ | … | $Y_i^{(s)}$ | … | $Y_n^{(s)}$ |

**Figura 5.13**

L'ipotesi di fondo è che, assunte le $\mathbf{A_i}$ indipendenti, le *n* coppie $(X_i,Y_i)$ siano indipendenti e identicamente distribuite. In generale, definita la variabile $Z_i$ $=(X_i,Y_i)$ ciascuna statistica può essere espressa come una funzione di Z, ovvero:

$$T= v(Z_1Z_2,……, Z_n)$$

Ciascun campione Bootstrap è realizzato effettuando il campionamento con rimessa di *n* elementi dell'insieme $\mathbf{Z}=\{Z_1,Z_2,.....,Z_n\}$. In questo modo, si costituisce il campione $\mathbf{Z^*}=\{Z^*_1,Z^*_2,.....,Z^*_n\}$, sulla base del quale è possibile calcolare la statistica $T^*=v(Z^*_1,Z^*_2,.....,Z^*_n)$. Il campionamento Bootstrap è effettuato tra i run e non all'interno del singolo in quanto in questo modo si tiene in considerazione la fonte ulteriore di variabilità connessa ai diversi input. Per effettuare la stima della distribuzione empirica di ciascuna statistica si estraggono *B* campioni Bootstrap e si considerano le statistiche ordinate $T_{(1)}^*,.....,T_{(B)}^*$. Il quantile α stimato risulta, in tal modo, pari a $T^*_{(\lfloor B\alpha \rfloor)}$. Nel caso di *s*=1, si verifica che alcune statistiche non possono essere applicate. In particolare, è possibile considerare solo le statistiche $T_1$, $T_2$ e $T_3$. Per la statistica $T_1$, Kleijnen, Cheng e Bettonvil, avendo esaminato un sistema di tipo M/M/1, hanno confrontato il valore tabulato della Z-Fisher per un errore di prima specie α noto, con il valore ottenuto in seguito alla trasformazione logaritmica per la normalizzazione dei dati $Y_i$ e $X_i$. Dovendo analizzare un modello più complesso, ovvero un sistema produttivo assimilabile a un sistema a reti di code, non è possibile conoscere la distribuzione dei dati in output. Per questo motivo, la normalizzazione è stata effettuata in relazione agli output ottenuti impiegando specifiche tecniche a seconda delle particolari condizioni. Anche in questo caso, dopo aver effettuato la normalizzazione, si confronta il valore ottenuto con quello tabellato senza effettuare il Bootstrap. Per $T_2$ e $T_3$, invece, si segue la procedura impiegata da Kleijnen, Cheng e Bettonvil. In altri termini, si costruisce un intervallo di confidenza bilaterale sulla base dei quantili stimati 1-α/2 e α/2 della distribuzione Bootstrap e si verifica se in esso è contenuto il valore di target della statistica. Nel caso di *s* pari a due, invece, non si considera $Z_i=(X_i,Y_i)$ ma $Z_i=(Y_i^{(1)}, Y_i^{(2)})$. In questo modo, si calcolano gli intervalli di confidenza, considerando, per la disuguaglianza di Bonferroni, non più 1-α come intervallo ma 1-α/2 dalla distribuzione Bootstrap di ciascuna statistica, realizzando il campionamento Bootstrap a partire dalle $Z_i$ sotto l'ipotesi nulla. Per valutare la validità del modello, si considera i due valori di ciascuna statistica a partire dal campione originario, ovvero $T=v((X_1, Y_1^{(r)}),...., (X_n, Y_n^{(r)}))$ con *r=1,2*. L'ipotesi nulla è

rigettata se almeno una delle due determinazioni della statistica ottenute dal campione originario risulta al di fuori dell'intervallo di confidenza. Naturalmente, almeno una statistica può essere inteso come la determinazione che assume valore massimo. Infatti, se è valida l'ipotesi nulla, allora **X**, $\mathbf{Y^{(1)}}$ e $\mathbf{Y^{(2)}}$ appartengono alla stessa popolazione. Di conseguenza, gli scarti tra i valori assunti dalle variabili reali e quelli simulati sia nella prima che nella seconda replicazione sono dovuti al campionamento casuale. Tuttavia, procedendo in tal modo non è più possibile confrontare l'intervallo di confidenza con i valori di target in quanto tale confronto non fornirebbe nessuna informazione tale da permettere di rigettare l'ipotesi nulla. Considerando, invece, se le statistiche calcolate dal campione originario appartengono all'intervallo, si determina se rigettare o meno l'ipotesi nulla. Infatti, se almeno una delle due determinazioni non rientra nell'intervallo si può affermare che le variabili reali e quelle simulate non appartengono alla stessa popolazione, ovvero gli scostamenti tra i valori assunti non sono solo fluttuazioni aleatorie, ma errori dovuti alla non validità del modello simulato. Il caso di *s* superiore a due può essere ricondotto a quello precedente attraverso un campionamento preliminare dei risultati della simulazione. In particolare, si possono considerare due strategie di campionamento:

a) *Campionamento di tipo conditioning*;
b) *Campionamenti di tipo non-conditioning*.

Nel primo caso, per ogni colonna *i-esima* della tabella si campionano due osservazioni $Y_i^{(r)}$ e $Y_i^{(r')}$ con r'≠r. La condizione r'≠r è dovuta al fatto che nel campione originario la probabilità di ottenere una coppia con valori identici è nulla nel caso di X e Y continui. Reiterando il procedimento *n* volte, si ottiene un campione Bootstrap a partire dal quale calcolare il valore della statistica T*.

Il campionamento *non-conditioning*, invece, si basa sull'assunzione che le variabili $A_i$ sono i.i.d. In questo caso, si ricampionano *n* coppie dall'intera tabella. In altri termini, si campiona il primo elemento dall'insieme degli *s* x *n* valori di **Y** e il secondo dagli *s* x *n-1* valori, ovvero il campionamento è effettuato senza reimmissione. Questa coppia di valori costituisce la prima coppia Bootstrap. La seconda coppia è ottenuta reiterando il procedimento dopo aver rimpiazzato la coppia precedente. Dopo *n* iterazioni, si ottiene il campione Bootstrap da cui

calcolare la statistica T*. Kleijnen, Cheng e Bettonvil hanno confrontato i due approcci per la statistica $T_2$, dimostrando che in entrambi gli approcci forniscono valori attesi di tutte le differenze tra le risposte delle simulazioni replicate sono nulle. Tuttavia, le loro varianze sono minori nel caso del primo approccio, quindi ci si aspetta che questa tecnica fornisca tests con potenze maggiori. Come nel caso di *s* pari a due, si rigetta l'ipotesi nulla se almeno uno delle *s* determinazioni della statistica T a partire dal campione originario è esterno all'intervallo di confidenza, calcolato considerando 1-α/s.

In genere, nell'ambito degli intervalli di confidenza il valore di *B* considerato è circa pari a 1000. In realtà, è possibile considerare il valore minimo di *B* necessario nel caso di s=1 sulla base della seguente relazione:

$$B_{min}=(2/\alpha)-1 \qquad (N)$$

Infatti, non sono necessarie tutte le informazioni concernenti la distribuzione g della statistica T*, ma solo i suoi α/2 e 1- α/2 quantili. Per stimare la distribuzione g, si considerano B osservazioni di T* che forniscono il campione ordinato $T^*_{(1)}$, $T^*_{(2)},\ldots,T^*_{(B)}$.. Di conseguenza il campione $g(T^*_{(1)})$, $g(T^*_{(2)}),\ldots,g(T^*_{(B)})$ è un campione ordinato estratto da una distribuzione uniforme su [0,1). Dalla teoria relativa alle statistiche ordinate, si verifica che:

$$f_{(i)}(u)=\frac{B!}{(i-1)!(B-i)!}f_U(u)[F_U(u)]^{i-1}[1-F_U(u)]^{B-i}=\frac{B!}{(i-1)!(B-i)!}u^{i-1}(1-u)^{n-i}$$

Quindi, si evince che la funzione densità di probabilità dell'i-esima statistica ordinata è di tipo Beta:

$$\text{Beta}(x)=(\Gamma(a+b)/\ \Gamma(a)\Gamma(b))x^{a-1}(1-x)^{b-1} \text{ con } a=i \text{ e } b=B-i+1.$$

Di conseguenza, il valore atteso $E(U_{(i)})=i/(B+1)$.

La stima del quantile α/2 può essere condotta considerando il valore di *i* che massimizza la funzione densità in α/2. Lo stesso procedimento può essere effettuato per 1- α/2. Si ottiene, quindi, che i due qunatili sono determinati rispettivamente da $T^*_{(1)}$ e $T^*_{(B)}$.. Poiché, $T^*_{(1)}$ è tale da soddisfare la seguente relazione:

$$T^*_{(1)}:\ g(T^*_{(1)})=\alpha/2$$

Si verifica che $E(g(T^*_{(1)}))=1/(B+1)=\alpha/2$. Da tale considerazione si ottiene la relazione (N). Naturalmente, sarebbe stato possibile ottenere lo stesso risultato considerando $T^*_{(B)}$.. Se s è maggiore di uno, allora è possibile applicare la

disuguaglianza di Bonferroni, quindi α è rimpiazzato da α/s. Tuttavia, se si estrae in maniera casuale un singolo valore per la statistica tra gli s disponibili dal campione originario, allora è necessario considerare comunque il caso di s=1.

# 6. Modello di un impianto per la produzione di congelatori

## 6.1. Introduzione

Alla luce di quanto illustrato precedentemente in via teorica, la simulazione ben si presta ad un'analisi di possibili scenari futuri. Lo studio ha riguardato, inizialmente, la formulazione di un modello logico-matematico rappresentativo della realtà aziendale, attuale e futura, in un linguaggio comprensibile al simulatore scelto. Il modello di simulazione ideato presenta caratteristiche di flessibilità ai cambiamenti relativi al mix produttivo, alla tipologia di modelli da produrre e ai tempi di processamento. La flessibilità deriva dall'implementazione delle logiche di funzionamento del modello mediante la programmazione in *Visual Basic* integrata con il simulatore ARENA e da dati di input forniti tramite interfacciamento al *database* aziendale. Sono stati condotti, in seguito, una serie di esperimenti per verificare le logiche del modello e per validarne le ipotesi mediante il confronto con i dati reali.

## 6.2. Formulazione del modello logico di simulazione

### *6.2.1. Le ipotesi del modello*

Il modello è stato realizzato seguendo regole di tipo logico, caratterizzate dalla presenza di blocchi connessi e moduli. La connessione dei blocchi è effettuata mediante semplici connettori, rappresentati graficamente da linee o frecce, o attraverso informazioni contenute all'interno dei blocchi stessi che consentono di inviare le entità ad un qualunque blocco del modello. Per ciascun blocco, comunque, esiste un relativo modulo dati che ne contiene tutte le informazioni necessarie come tempi di processamento, costi, distanze, distribuzioni di probabilità.

Il numero e l'entità delle ipotesi cui è soggetto il modello devono essere scelti in maniera appropriata cercando di realizzare un buon compromesso tra la

necessità di realizzare un modello aderente alla realtà e quella di non appesantirlo con particolari insignificanti per il livello di precisione richiesto.

Ciascun modello è contraddistinto da ipotesi *generali* o *di sistema* e da ipotesi *particolari*.

Nel passaggio dal sistema reale al modello logico si sono adottate le seguenti ipotesi *generali*:

- Tempi di trasporto interno trascurabili;
- Fermi macchina assenti, eccetto per l'impianto di schiumatura;
- Le stazioni in linea non prevedono buffer interoperazionali;
- Pannellatrici fuori linea.

a) *Tempi di trasporto interno trascurabili*: il trasporto delle materie prime presso le stazioni dove saranno lavorate è affidato a due *carrellisti*. Per quel che riguarda il trasferimento dei semilavorati, invece, questi avanzano lungo la linea di produzione mediante sistemi a rulli. In entrambi i casi, l'ingombro, così come il peso dei materiali da movimentare, non è eccessivo, quindi, tenendo conto anche delle esigue distanze da percorrere, si possono ritenere trascurabili i tempi di carico, movimentazione e scarico o, al limite, li si possono inglobare nei tempi di lavorazione.
b) *Fermi macchina assenti*: i fermi macchina, escludendo quelli dovuti allo sbilanciamento della linea di produzione che sono oggetto del nostro studio, sono dovuti principalmente a quattro cause:

   - *Mancanza di materie prime*. I fermi macchina causati dalla mancanza di materie prime sono esigui: i buffer di materie prime alle stazioni di lavoro vengono riforniti da carrellisti, sotto la supervisione del responsabile dei magazzini. Analisi condotte all'interno dello stabilimento hanno permesso di verificare un ottimo livello di riordino dei buffer. Attraverso i colloqui con i responsabili delle varie stazioni e da dati storici è, inoltre, emersa la sostanziale assenza di guasti generalizzati alle linee.
   - *Guasti*. Guasti di consistente entità alle macchine[3], hanno probabilità di accadimento molto bassa, quindi è lecito, nella costruzione del modello non tenerli in conto. Non è altrettanto significativo tenere in conto i guasti di limitata entità la cui frequenza di accadimento è piuttosto alta ma che sono caratterizzati da tempi di ripristino brevissimi.
   - *Interventi di manutenzione*. Gli interventi di manutenzione avvengono in maniera programmata. Le macchine che fanno parte del modello saranno oggetto di operazioni di manutenzione

[3] Quelli, vale a dire, che prevedono un tempo di ripristino particolarmente lungo e il conseguente fermo della produzione.

durante le ore in cui la linea produttiva è ferma, ossia nel turno notturno o durante il fine settimana.

- *Set-up*. L'ipotesi di ritenere i tempi di attrezzaggio trascurabili è forte e potrebbe inficiare i risultati ottenuti tramite la simulazione. Per questo motivo, tale ipotesi non è stata tenuta in considerazione nella realizzazione della fase di schiumatura, in cui vengono effettuati set-up lunghi e all'interno del turno lavorativo. Inoltre, quando si producono diversi modelli appartenenti alla stessa linea di produzione, non è possibile trascurare il set-up breve.

c) *Le stazioni in linea non prevedono buffer interoperazionali*: i buffer interoperazionali non hanno, nell'azienda oggetto dello studio, nessun tipo di pianificazione. Piccolo accumuli di materiale si creano, tuttavia, a valle e a monte di quasi tutte le stazioni e sono per lo più dovuti allo sbilanciamento del flusso produttivo. La loro presenza, comunque, non altera i risultati ottenuti da una simulazione che non ne prevede l'esistenza.
d) *Pannellartrici fuori dai limiti del modello***:** le analisi condotto sulla linea hanno consentito di verificare come sia necessario il ricorso a buffer di disaccoppiamento per tale postazione. Per questo motivo tale postazione fuoriesce dai confini del modello logico elaborato.

### *6.2.2. Il modello logico*

Data la complessità del modello da realizzare e l'elevato numero di "blocchi" da utilizzare ci si è avvalsi di sottomodelli che permettono all'utilizzatore di avere maggior controllo sul modello ed una visione più snella dei vari reparti.

Si è ottenuto un elevato grado di flessibilità del modello grazie all'interfacciamento con un database contenente le informazioni relative a tutte le tipologie di prodotti realizzati in azienda, quali tempi di processamento, relazioni per i set-up ecc, nonché il piano di produzione.

Il modello completo dello stabilimento produttivo è così schematizzabile:

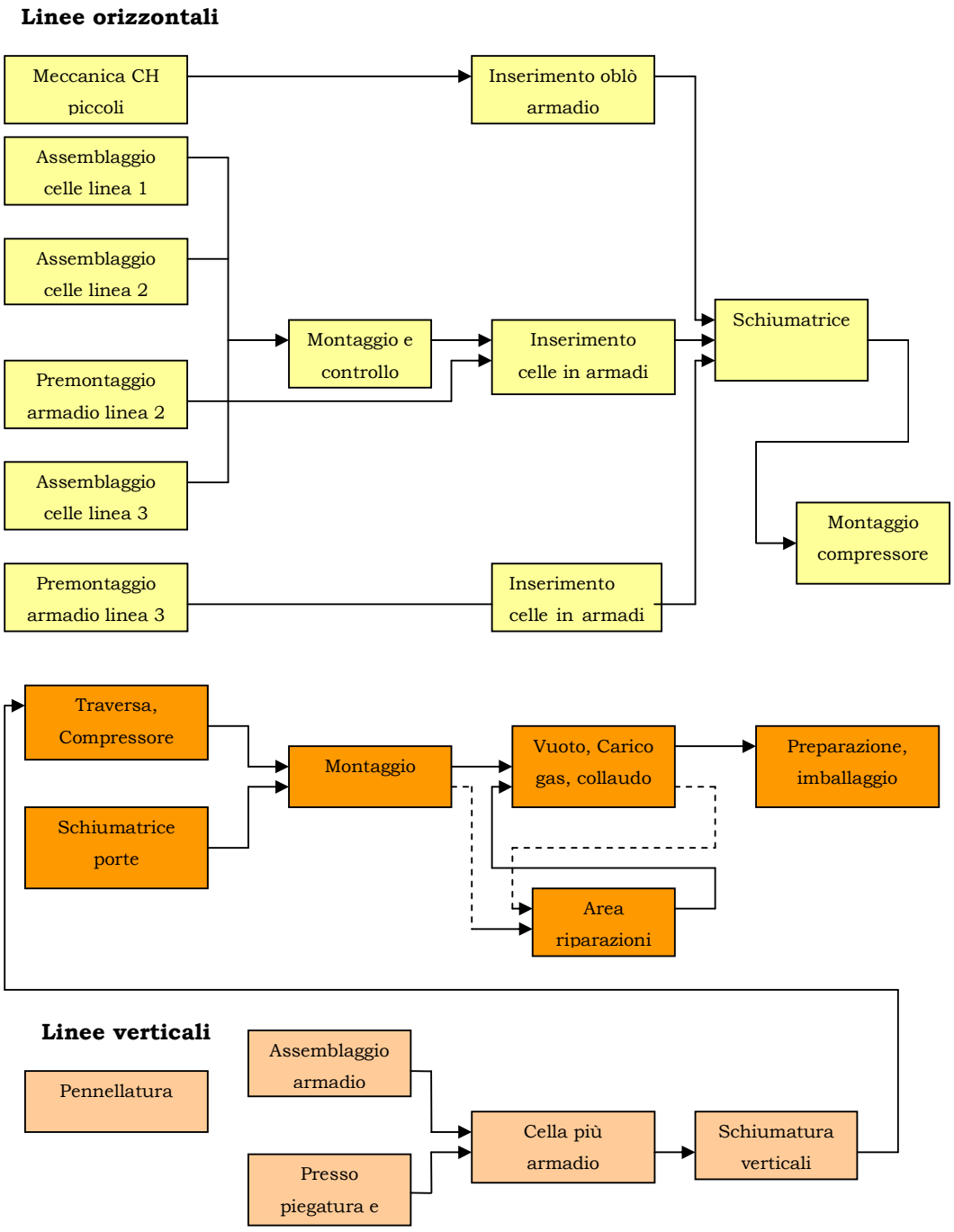



**Figura 6.1 Schema del modello logico**

Di seguito è presentata l'analisi dei singoli sottomodelli seguendo il flusso logico da implementare.

## Inizializzazione del modello

La pannellatrice produce principalmente per ripristinare i buffer che si trovano a bordo macchina, i quali fungono da scorte polmone per la linea di produzione. Per tale motivo è stata considerata in separata sede. È stata schematizzata mediante tre blocchi Create, tre blocchi VBA e tre blocchi Dispose come mostrato nella Figura 6.2.

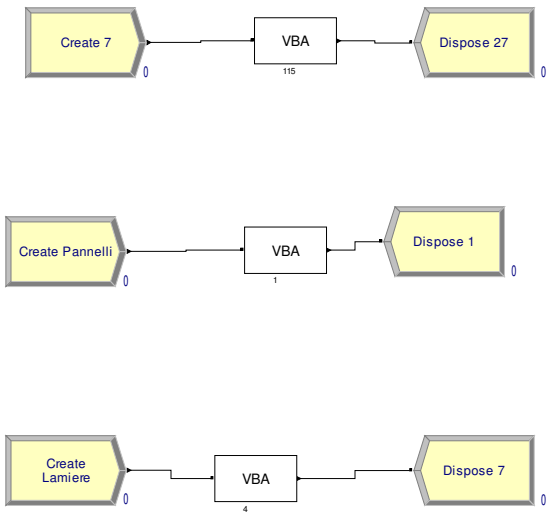



**Figura 6.2 Modello logico rappresentativo dell'inizializzazione**

Il blocco Create ha la funzione di generare entità: in questo caso ciascuno genera una sola entità che ha lo scopo di inizializzare la procedura elaborata in Visual Basic, contenuta nel blocco VBA. Queste entità non entreranno a far parte del processo produttivo ma verranno rimosse tramite i blocchi Dispose. Il blocco VBA è l'unico, nell'ambito di ARENA, che consente di programmare in Visual Basic allo scopo di rendere più flessibile il funzionamento del modello che si vuole realizzare. Nell'inizializzazione il VBA è stato utilizzato al fine di collegarsi al database, con il quale interagisce mediante linguaggio *SQL*, e di generare tante entità quante indicato nel piano di produzione. A ciascuna entità creata vengono assegnati, in base alle informazioni contenute nel database, una serie di attributi, quali modello, maschere dedicate, tempi di processamento e linea di produzione cui deve essere destinata. In particolare, il primo VBA genera i pannelli dei congelatori orizzontali e verticali, mentre il secondo genera le lamiere.

Ogni linea di produzione inizia con un *element* **Assign**, collegato virtualmente ai VBA tramite un'istruzione che invia le entità create all'Assign della linea dedicata.

**Premontaggio Armadio**

Le linee del premontaggio sono state modellate in maniera dettagliata differenziando opportunamente le cinque linee di produzione.

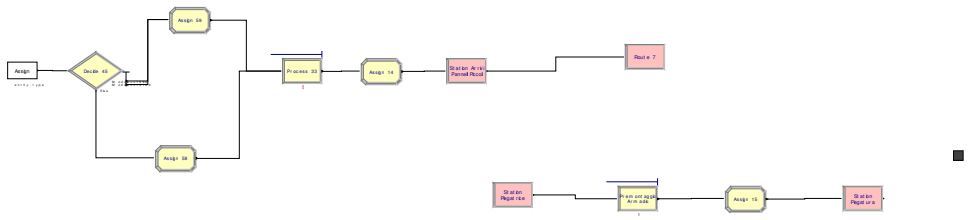


**Figura 6.3 Modello logico per la linea 1 del pre-montaggio armadio**

L'element Assign rappresenta il punto di ingresso delle *entità* all'interno del modello. Come accennato in precedenza ciascun operatore all'inizio di ogni turno ha a disposizione un buffer di pannelli prelavorati durante il turno precedente, quindi l'entità uscente dal modulo è un unico pannello che andrà a formare l'armadio del modello *piccolo*. Il tempo di premontaggio armadio a sua volta è dato dalla somma di due aliquote: il tempo necessario alla macchina piegatrice per piegare il pannello e il tempo necessario all'operatore per effettuare le operazioni di foratura, montaggio polionda, etc. Il modulo **Process** indica l'elaborazione di un'*entità*. Contiene le opzioni per prendere e rilasciare una o più entità, un tempo di processamento (*Process Time)* associato all'intera operazione, una o più risorse. Nel nostro caso il tempo di processamento è stato definito e le risorse sono rappresentate da un macchina (piegatrice) e un operatore.

A questo punto *l'entità,* che in entrata era un pannello, all'uscita del modulo **Process** avrà subito la trasformazione in *armadio piccolo*. Per rappresentare tale variazione si utilizza un modulo **Assign** dove viene assegnata una nuova *Entity Type*, che formalizza il cambiamento logico avvenuto e una nuova *Entity Pictures* per mostrare anche visivamente tale variazione.

I collegamenti logici tra i blocchi viengono realizzati attraverso i blocchi **Advance Transfer**.

A questo punto l'armadio viene inviato, grazie ad un connettore logico, al sottomodello "Inserimento Oblò in Armadio", che verrà analizzato in seguito.

Per le cinque linee il premontaggio la logica è equivalente, eccetto per alcuni vincoli precedentemente descritti, a quella della linea 1 anche se con alcune differenze di lavorazione.

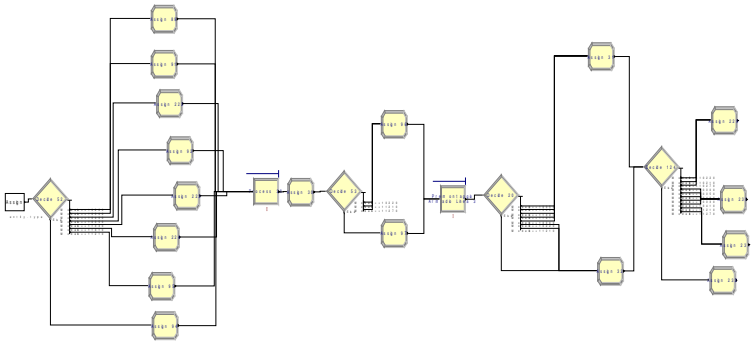


**Figura 6.4 Modello logico per la linea 3 del pre-montaggio armadio**

Il modulo **Decide** differenzia i pannelli medi da quelli grandi e li invia in due moduli Assign differenti, in cui viene definita la nuova *Entity Type*, rappresentata dall' armadio medio o l'armadio grande, ed una nuova *Entity Picture* per visualizzare la lavorazione realizzata. Le nuove entità create vengono quindi inviate nel sottomodello *"Inserimento Celle in Armadio Linea 3"* .

**Assemblaggio celle**

L' operazione di assemblaggio celle presenta differenze tra la linea 1 e le due restanti.

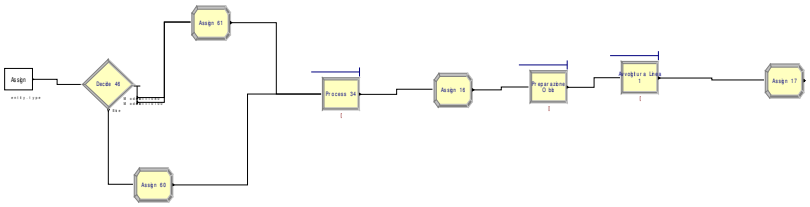


**Figura 6.5 Modello logico per la linea 1 dell'assemblaggio celle**

tato il modello logico della linea tre dell'assemblaggio celle.

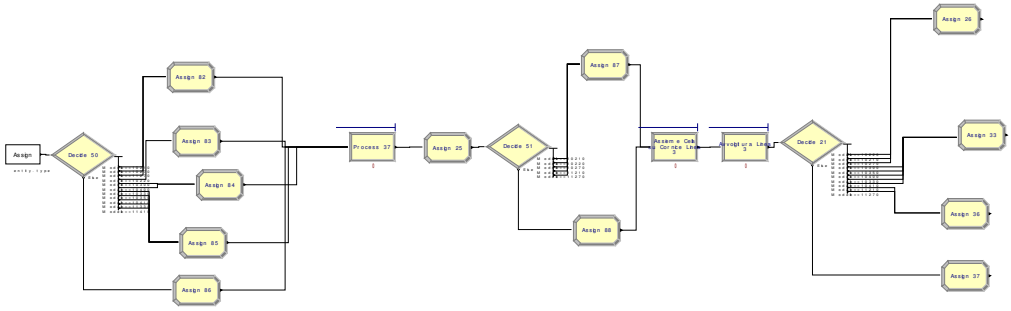


**Figura 6.6 Modello logico per la linea 3 dell'assemblaggio celle**

Il modulo **Decide** differenzia le celle prodotte in celle medie modello monovasca, celle medie modello doppiavasca, celle grandi modello monovasca, celle grandi modello doppiavasca e le invia nei rispettivi moduli **Assign** dove vengono assegnati i tempi necessari ad effettuare il montaggio e il controllo dello scambiatore.

Le linee dedicate ai verticali sono schematizzate in maniera diversa perchè il processo non prevede l'operazione di avvolgitura.

## Montaggio e controllo scambiatore

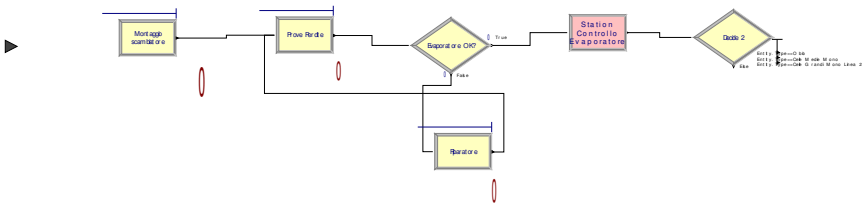


**Figura 6.7 Modello logico del montaggio e controllo scambiatore**

In questo sottomodello arrivano le celle dalle tre linee di produzione. In due postazioni, presiedute da due operatori e schematizzati mediante due moduli **Process,** si effettuano il montaggio e il controllo dello scambiatore. Gli scarti, sono molto limitati e comunque riparabili. Attraverso un modulo **Decide,** si effettua un controllo sullo scambiatore, con una percentuale di scarti pari al 0,5% che vengono inviati in un modulo **Process,** in cui viene effettuata la riparazione della cella dallo stesso operatore addetto al montaggio scambiatore. In uscita, le celle vengono indirizzate alle postazioni di inserimento nell'armadio: è il modulo

**Decide** che indirizza opportunamente le celle in base alla loro dimensione e alla loro tipologia.

**Assemblaggio cella ed armadio**

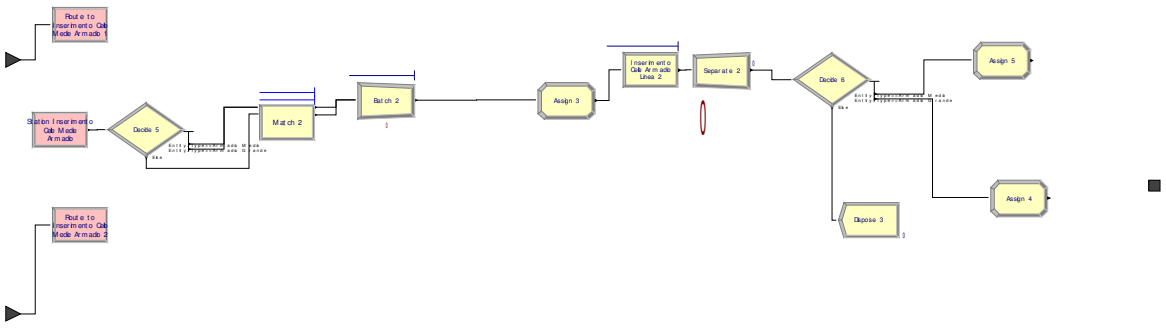



**Figura 6.8 Modello logico rappresentativo dell'inserimento delle celle nell'armadio**

Le entità in entrata in questo sottomodello sono l'armadio proveniente dal sottomodello "assemblaggio armadio", e le celle provenienti dal montaggio e controllo dello scambiatore. L'inserimento della cella nell'armadio è stato modellato attraverso il costrutto logico **Match-Batch-Process**. Il blocco **Match** pone in attesa in code diverse un numero specificato di entità differenti (in questo caso due). Quando in entrambe le code sarà presente almeno un'entità, queste lasceranno il blocco e raggiungeranno il modulo **Batch** che permette di raggrupparle in un'unica entità gruppo. Questo costrutto rappresenta l'operazione di assemblaggio dal punto di vista logico, cioè a tempo zero. Per modellare anche tempisticamente l'operazione, si utilizza il blocco **Process** preceduto dal modulo **Assign**, nel quale si assegna il tempo necessario ad effettuare l'operazione. Prima che le nuove *entità* create lascino il sottomodello, ancora un modulo **Assign** assegna una nuova *Entity Type (Armadio L1)* che verrà richiamata nel sottomodello successivo, e una nuova *Entity Picture* per l'animazione grafica. La rappresentazione del modello logico relativo alle linee due e tre e alle linee verticali, risulta equivalente a quello della linea uno. A questo punto il prodotto è pronto per subire la schiumatura.

**Schiumatura Armadi orizzontali**

Gli armadi provenienti dalle linee di premontaggio attendono in coda ai rispettivi forni, uno per ogni linea, l'arrivo del carrello per l'operazione di carico. L'impianto di schiumatura è formato da sei maschere, su ognuna delle quali è montato un particolare stampo, in funzione del mix produttivo del turno. Le sei maschere sono alimentate mediante un solo carrello di capacità limitata (può trasportare un solo armadio per volta), il quale si occupa, muovendosi lungo un percorso guidato, di prelevare gli armadi ai forni, portarli alle rispettive maschere, effettuare le operazioni di scarico armadio "da schiumare"/carico armadio "schiumato", e quindi portare questi ultimi alla linea di montaggio. Il tutto avviene in modo automatizzato. In condizioni di regime sono attive tutte e sei le maschere, la prima che raggiunge il limite di estrazione prenota il carrello. Affinché quest'ultimo possa caricare il pezzo in attesa alla maschera per portarlo alla linea di montaggio, è necessario che sia pronto, ai forni di preriscaldamento, un armadio "da schiumare" dello stesso tipo di quello "schiumato" che ha richiesto il carrello. Fino a quando questa condizione non si verifica il congelatore "schiumato" resta in attesa nella maschera, mentre il carrello rimane disponibile nel caso in cui anche altre maschere raggiungano il limite di estrazione e magari siano presenti ai forni armadi dello stesso tipo di quelli in esse contenuti.
Per modellare questo particolare sistema si sono, al solito, adottate una serie di ipotesi semplificative:

- La variabile *Tempo Schiumatura* è pari, per ogni modello, alla somma del tempo di iniezione schiuma più il limite d'estrazione;
- All'inizio del turno di lavoro, i maschi per la schiumatura sono già alla temperatura giusta. L'ipotesi può essere ritenuta valida visto che i maschi vengono riscaldati nel turno notturno in cui le maschere sono inattive.

La modellazione dell'intero impianto è rappresentata in Figura 6.9.

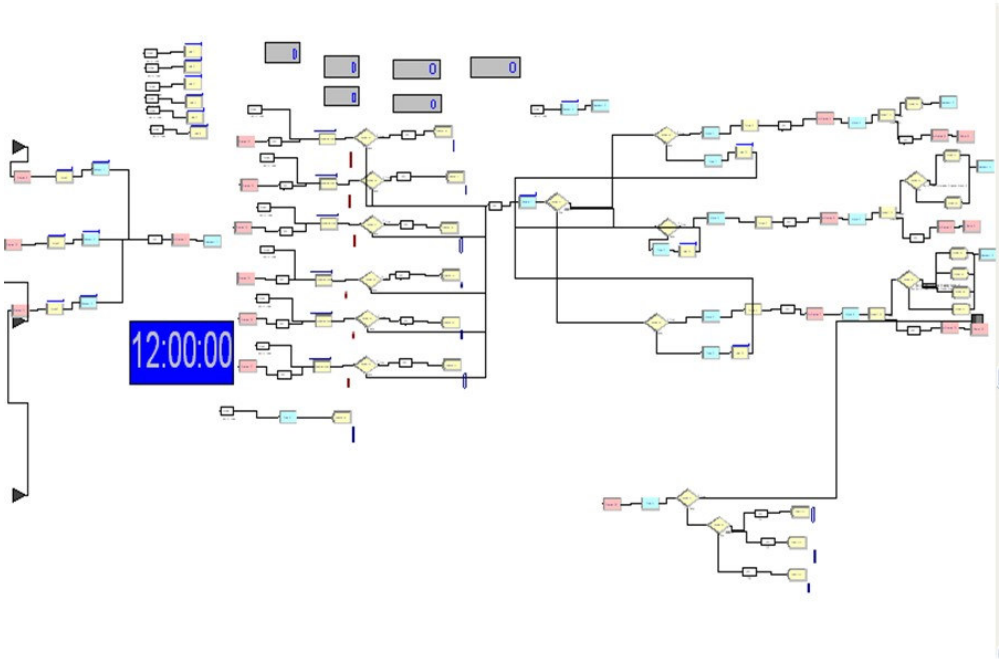



**Figura 6.9 Reparto di schiumatura**

Dopo che gli armadi e le celle sono stati assemblati nel reparto precedente, l'entità raggiunge la propria stazione di riscaldamento ai forni modellata, fisicamente e logicamente, tramite il blocco **Station** ed entra nel blocco **Hold**. Questo modulo trattiene una o più *entità* in coda fino a quando viene emesso un segnale, o fino a quando una specificata condizione diviene vera (*scan*), o indefinitamente; in quest'ultima circostanza le *entità* possono essere rimosse dalla coda solo mediante opportuni moduli. Nel modello realizzato, il forno di una qualunque linea trattiene le entità in coda in base a condizioni espresse da una funzione (*user function*) che viene richiamata ogni qual volta l'entità passa nel forno. La funzione esegue contemporaneamente un controllo sull'attributo "modello" delle entità, sul numero di maschere dedicate a ciascuna linea e lascia passare tante entità quante sono le maschere dedicate alla linea in quel momento non utilizzate. Ciò avviene all'inizio di ciascun turno lavorativo e dopo ogni set-up. Il controllo delle maschere dedicate avviene, in realtà, attraverso l'interrogazione di sei entità, ciascuna in coda ad un blocco Hold esterno al processo di schiumatura. Queste entità vengono inviate in coda agli Hold dal VBA

iniziale che, inoltre, assegna l'attributo modello a ciascuna di esse[4]. Le entità che verificano la funzione vengono rilasciate ed entrano nel blocco **Request** in attesa del carrello che le porti alle schiumatrici. Prima di arrivare alle schiumatici, le entità passano per un altro blocco VBA all'interno del quale si impongono delle condizioni di indirizzamento del carrello verso la schiumatrice che lo ha richiamato. Il trasferimento fisico del carrello avviene attraverso il blocco **PickStation**. Dopo il passaggio attraverso il modulo **Station** necessario alla locazione fisica e logica delle maschere, le entità passano per un altro VBA che ha la funzione di assegnare l'attributo che individua la maschera che le processerà. Contemporaneamente lo stato della maschera passa da non utilizzata a utilizzata, annullando la funzione che regola il passaggio ai forni. Le entità arrivano, quindi, al modulo **Process** rappresentativo della maschera di schiumatura. Dopo le schiumatrici è presente un VBA che conta il numero di congelatori schiumati su ogni linea ed esegue il set-up lungo qualora tale numero verificasse la seguente relazione:

$$Q_{L_i} = \text{Lotto} - (M-1)\cdot UE \qquad \text{per } i=1,2,3$$

dove:

- $Q_{L_i}$ è la quantità ancora da produrre sulla linea i-esima in corrispondenza della quale eseguire il set- up lungo;
- M è il numero di maschere dedicate alla linea;
- *UE* è il numero di unità equivalenti, cioè il numero di congelatori schiumati con una maschera, in un turno lavorativo. Esso cambia da modello a modello in base al tempo di schiumatura.

Contemporaneamente, il VBA crea un'entità che rappresenta la squadra di set-up e la invia alla schiumatrice. Lo stesso VBA ha la funzione di effettuare il set-up breve qualora su di una linea si produca più di un modello.
La quantità in corrispondenza della quale avviene il set-up breve dipende dal numero di maschere dedicate e da un parametro n calcolato come rapporto tra tempo di set-up e tempo di schiumatura. Questa quantità viene calcolata mediante un "*ciclo for*" e quando verifica l'uguaglianza con la quantità ancora da

[4] Ad esempio, se le entità in coda ai primi tre Hold hanno l'attributo modello uguale a CH80 lavorato sulla prima linea, allora il numero di maschere dedicate a quella linea è proprio pari a tre.

produrre, rappresentata da una variabile che si aggiorna ogni qualvolta passa l'entità schiumata nel VBA, si esegue il set-up breve.

I **Decide** collocati dopo ogni schiumatrice hanno la funzione di separare l'entità congelatore schiumato dall'entità set-up che, prima di lasciare la linea di produzione, entra in un blocco VBA nel quale si modifica lo stato della risorsa schiumatrice da *utilizzata* a *non utilizzata*. In questo modo la *user function* risulta verificata e si ha il passaggio di un'entità al forno.

La schiumatrice che ha terminato la schiumatura di un congelatore richiama il carrello che esegue lo scarico solo quando è presente un congelatore dello stesso tipo al forno. Il carrello è movimentato tramite un modulo **Move** che lo invia al forno di preriscaldamento. Per caricare l'armadio "da schiumare" sul carrello è stato usato il modulo **Pickup**. Il modulo in questione rimuove un certo numero di *entità* consecutive da una coda, cominciando da quella che occupa una specifica posizione nella stessa. Queste *entità* si muoveranno, nel resto del modello, insieme all'*entità* che le ha rimosse, fino a quando non sarà fornito un comando logico idoneo a separarle. Nel nostro esempio, l'*entità* che entra nel modulo è unica (l'*entità* che controlla il carrello), ed è unica anche l'*entità* da rimuovere dalla coda (primo armadio pronto nella coda ai forni). Ancora, un modulo **Move** è stato usato per inviare sia il carrello che l'armadio alle maschere di schiumatura. Lo scambio che avviene presso la schiumatura, tra armadio schiumato e quello da schiumare, viene realizzato tramite il blocco **Dropoff**, il quale permette di rimuovere un certo numero di *entità* da un gruppo e di inviarle in altri moduli secondo quanto specificato dai connettori grafici. L'*entità* che fuoriesce dal ramo *original* del modulo **Dropoff** rappresenta il congelatore schiumato e, tramite il modulo **Transporter**, raggiunge il blocco **Station** che rappresenta la linea di montaggio. A questo punto l'entità libera il carrello tramite il modulo **Free** e, prima di abbandonare il sottomodello, un modulo **Decide** effettua un controllo qualità sul prodotto. La percentuale dei pezzi che non superano il controllo è molto bassa, ma il prodotto risulta irreparabile così che il modulo Dispose conterà gli scarti del reparto schiumatura.

L'entità che fuoriesce dal ramo members del modulo Dropoff, essendo l'armadio da schiumare, è inviato alle maschere di schiumatura della linea tramite una PickStation al Process schiumatrice che l'ha richiesta.

Il funzionamento logico della schiumatrice è identico per ogni linea di produzione.

**Montaggio compressore**

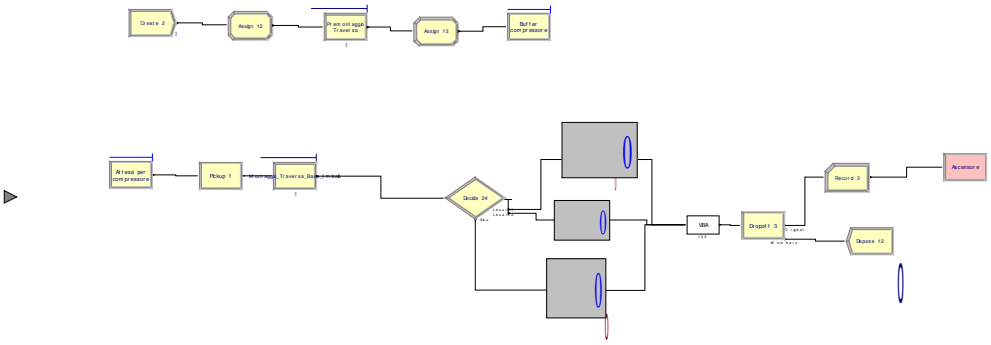


**Figura 6.10 Modello logico rappresentativo del reparto montaggio compressore**

La fase di assemblaggio del compressore, rappresenta un punto particolarmente importante per la programmazione della produzione nell'azienda oggetto dello studio. Il piano di Produzione dell'azienda fa riferimento, in particolar modo, proprio a tale fase, al termine della quale il congelatore è definito "*impostato*". Questa postazione si occupa del montaggio della traversa del compressore nell'apposito vano[5]. Il congelatore schiumato correttamente arriva al blocco **Hold** dove attende solo in caso di mancanza del compressore montato sulla traversa. L'*entità congelatore schiumato*, entra nel modulo **Pickup** e rimuove l'*entità compressore* dalla coda del Buffer compressore. Il modulo **Process** effettua il montaggio del compressore nel congelatore. Per rendere immediata la lettura dei congelatori prodotti alla fine del turno di lavoro, sono stati inseriti contatori numerici che visualizzano il numero di entità uscenti dal blocco **Process** relativo al montaggio dello scambiatore. Il VBA presente in questo sottomodello ha la funzione di generare una tabella in cui viene registrato il numero di congelatori impostato su ciascuna linea.

[5] Prima di questa operazione è previsto un operatore fuori linea addetto al premontaggio del compressore sulla traversa.

Il modulo **Record** viene utilizzato per registrare una serie di dati che rappresenteranno parte degli output del modello di simulazione.
A questo punto i congelatori, tramite nastro trasportatore, raggiungono l'ascensore e vengono trasferiti al piano superiore. L'ascensore è stato modellato con il modulo **Leave** del template **Advanced Transfer**.

**Montaggio Manuale**

Il sottomodello relativo al montaggio manuale racchiude in sé tutte le operazioni di montaggio di piccole parti che avvengono al piano superiore dello stabilimento. In particolare nel **Process** *Postazione Saldatura* si riscontra la confluenza delle linee orizzontali e verticali.

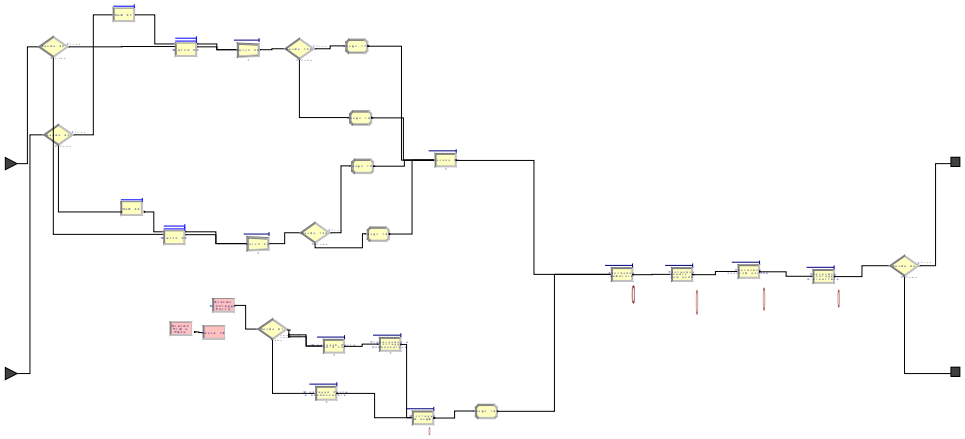


**Figura 6.11 Modello logico rappresentativo del reparto di montaggio**

I congelatori orizzontali arrivano alla **Station** *Primo Piano* dove viene eseguito il montaggio delle porte e dello scambiatore. Il modulo **Decide** differenzia i prodotti della linea uno da quelli delle altre due linee, per effettuare una diversa operazione di montaggio porta. Infatti, mentre per i modelli piccoli un solo operatore effettua sia il montaggio della porta che quello del condensatore, ed quindi è stato modellato con un solo **Process** con tempo di processamento pari alla somma dei tempi delle due lavorazioni , per i modelli delle linee 2 e 3 si hanno due postazioni di lavoro presiedute da altrettanti operatori che svolgono in serie le due lavorazioni. Sege, poi, un blocco **Process** nel quale viene montata la maniglia alla porta del congelatore.

Le linee verticali confluiscono nel **Process** *Montaggio porte e maniglie*. I successivi quattro **Process** sono comuni a tutte le linee, nel primo viene effettuata l'operazione di saldatura per la chiusura del circuito di refrigerazione; nel secondo viene montato il quadro di comando; nel terzo viene effettuato un controllo perdite e nel quinto viene montata la morsettiera per l'alimentazione.
In uscita dal sottomodello, con l'aiuto di un modulo **Decide**, si effettua un controllo qualità con percentuale di scarti del 3%.

**Area di rilavorazione**

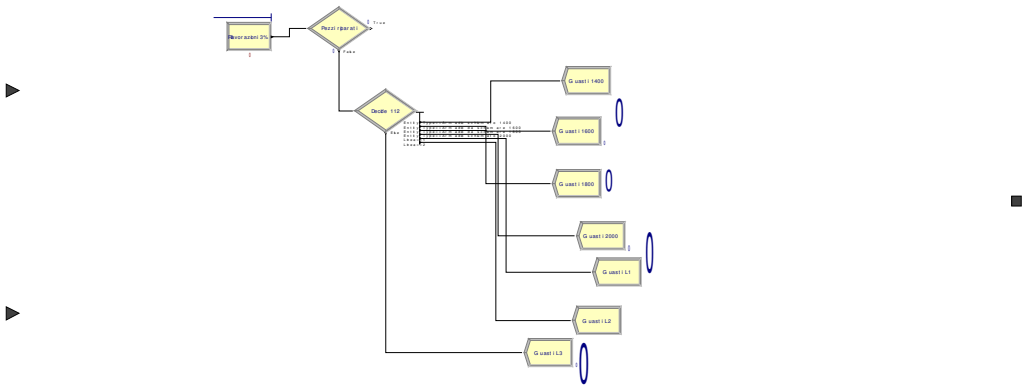


**Figura 6.12 Modello logico rappresentativo dell'area di rilavorazione**

All'area riparazioni arrivano le entità che non hanno superato il collaudo o altri controlli qualità. Essendo la percentuale dei prodotti scartati molto bassa, risulterebbe poco produttivo allocare un operatore addetto esclusivamente a tali attività. Per questo motivo la rilavorazione e la riparazione viene effettuata da operatori impegnati in altre attività. Il modulo **Process** relativo alla stazione di riparazione presenta quindi un tempo di processamento molto lungo. Dopo la rilavorazione i prodotti subiscono un ulteriore controllo, schematizzato con un blocco **Decide**, che viene superato dal 99% delle unità. Il restante 1% rappresenta il definitivo scarto di produzione da suddividere per le tre linee.

**Carico Gas e Collaudo**

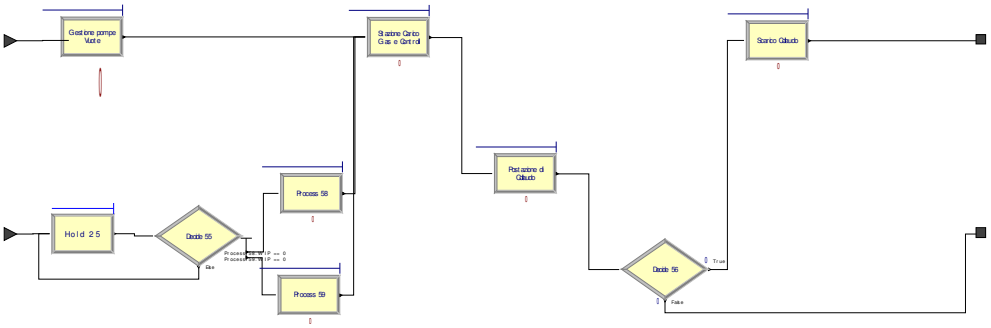



**Figura 6.13 Modello logico rappresentativo delle stazioni di vuoto, carico gas e collaudo**

Questo sottomodello presenta due entrate perché vi giungono sia i congelatori che hanno superato il controllo della saldatura e sono pronti al collaudo, sia quelli provenienti dall'area di riparazione. Le entità che hanno superato il controllo entrano nel primo blocco **Process** che rappresenta la stazione di creazione vuoto all'interno dei congelatori. Il blocco presenta un set di risorse che modella le tredici pompe disponibili ed un unico operatore che controlla l'esecuzione dell'operazione. Le entità che provengono dall'area di riparazione afferiscono ad una postazione mobile contenente due pompe aggiuntive. Il modulo **Hold** trattiene le entità fino a che una delle due pompe non risulti libera. Potrebbe capitare che in coda al modulo **Hold** ci siano più di due entità in attesa di essere processate e, quindi, una volta liberatasi una delle due pompe tutte le entità abbandonino il modulo. Poiché le pompe disponibili sono al massimo due, un blocco **Decide** effettua una verifica sullo stato dei due moduli **Process** e rimanda in coda all'**Hold** le entità in eccesso. Superata la fase di creazione del vuoto, le entità raggiungono la stazione dove un unico operatore carica il gas: ciò avviene all'interno di un blocco **Process.** A questo punto le entità sono pronte per il collaudo. L'operazione è modellata con un blocco **Process** all'interno del quale è situato un set 128 risorse rappresentanti le baie di collaudo. A valle delle postazioni di collaudo, esiste un nuovo controllo qualità che viene superato dal 95% dei pezzi che sono pronti per essere imballati e messi in magazzino. Il restante 5% viene inviato all'area rilavorazione.

**Preparazione, imballaggio e magazzino**

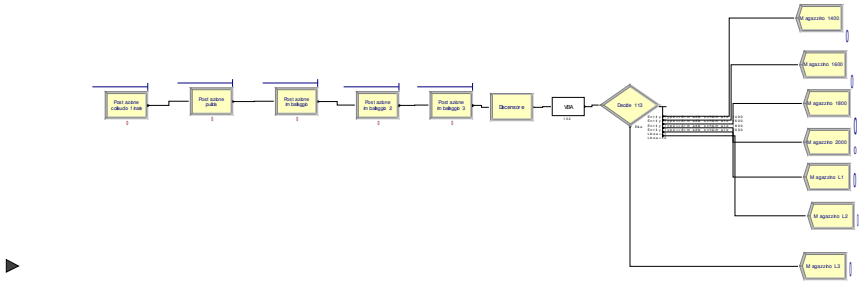


**Figura 6.14 Modello logico rappresentativo dell'area di preparazione ed imballaggio**

Le operazioni di preparazione finale del prodotto ed imballaggio sono state modellate con quattro blocchi **Process** presieduti da altrettanti operatori. Nel primo viene effettuato un controllo perdite dell'impianto e verificata l'accensione della ventola (se esistente) e della luce della porta. In seguito viene montata la mascherina laterale necessaria alla chiusura del vano compressore. Nel secondo modulo viene modellata la fase di pulizia del congelatore, del collaudo visivo e dell'inserimento dei cestelli di plastica nel congelatore. A seguire, nel terzo blocco **Process**, avviene l'inserimento nel congelatore degli arredi e di tutti gli accessori necessari. Nell'ultimo si procede, infine, al rivestimento con il cartone per l'imballaggio finale. Le entità in uscita dall'ultimo blocco attraversano un blocco **Delay** che rappresenta il tempo impiegato nella discesa dal piano superiore al magazzino che si trova al piano inferiore.
Il VBA presente in questo sottomodello ha la funzione di generare una tabella in cui vengono registrate le quantità giunte in magazzino alla fine del turno lavorativo.
Chiude la modellazione del processo produttivo l'area di magazzino dove un modulo **Decide** differenzia le entità in base alla linea di produzione di appartenenza.

## 6.3. Validazione del modello di simulazione

La verifica o debugging consiste nell'assicurare che il programma segua in modo corretto il flusso logico voluto, senza interruzioni inaspettate. La verifica deve essere prevista continuamente nella fase di creazione mediante simulazioni pilota.

Superata la fase di verifica del modello si passa alla sua validazione. Nella fase di validazione è necessario verificare se il modello realizzato fornisce risultati paragonabili al comportamento del sistema reale. Più in particolare si deve verificare se le misure di prestazione del sistema reale sono bene approssimate dalle misure generate dal modello di simulazione. Tale fase è preceduta da un'elaborazione dei dati reali, necessaria al confronto con gli output del modello. A partire dai reports di produzione per turno, relativi agli ultimi due anni di attività dello stabilimento, è stato selezionato il turno lavorativo caratterizzato da massima produttività su ciascuna linea di produzione. Come parametro di prestazione del sistema si è scelto il numero di congelatori impostati. Il motivo della scelta è da ricondursi al fatto che l'azienda misura la sua produttività a valle della fase di assemblaggio del compressore. Tale parametro è stato confrontato con gli output del modello ARENA, in particolare con la variabile *Montaggio_Traversa_Base_Imballo.NumberOut.* Dunque, per valutare la corrispondenza tra tempi di processamento inseriti nel modello di simulazione e quelli reali, si è deciso di simulare lo scenario caratterizzato dai seguenti parametri:

- *Lunghezza del turno:* turno centrale di 8 ore lavorative;
- *Mix produttivo:* (3,2,1);
- *Piano di produzione:* Figura 6.15
- *Manodopera*: a pieno regime e distribuita alle linee in modo standard.

| Modelli | Quantità | Linea | Note |
|---|---|---|---|
| CH100 | 110 | L1 | Con setup breve |
| CH130 | 75 | | |
| CH210 | 65 | L2 | Con setup breve |
| CH270 | 29 | | |
| CH330 | 36 | L3 | Con setup breve |
| CH410 | 14 | | |

**Figura 6.15 Esempio di piano di Produzione**

Si è deciso di produrre due modelli diversi di congelatore su ciascuna linea di produzione, in modo da realizzare un setup breve su ogni schiumatrice, e si è deciso, inoltre, di produrre i modelli caratterizzati dal massimo tempo di schiumatura.

Poiché gli input del modello di simulazione seguono delle distribuzione statistiche, gli output sono essi stessi variabili aleatorie. Non essendo nota la funzione di distribuzione della v.a. *Montaggio_Traversa_Base_Imballo.NumberOut* si ricorre al teorema del limite centrale per applicare le tecniche di stima intervallare. Il teorema del limite centrale afferma che la somma di n v.a. s-indipendenti ed equidistribuite con media $\mu$ e varianza $\sigma^2$ tende, per $n \to \infty$, ad essere distribuita come una v.a. Normale con media $n\mu$ e varianza $n\sigma^2$. In generale, nel caso in cui il campione casuale non è tratto da una popolazione Gaussiana, il teorema del limite centrale è applicabile per dimensioni campionarie n>30, anche se talvolta può risultare sufficiente una n>5 nel caso di variabili aleatorie con funzioni di distribuzione di probabilità caratterizzate da una certa simmetria [62].

Il primo passo da affrontare per stimare la misura di prestazione di interesse è quello di determinare la dimensione campionaria n in corrispondenza della quale è possibile affermare che la media campionaria è una v.a. Gaussiana con media $\mu$ e varianza $\sigma^2/n$. A tale proposito sono state eseguite 150 repliche della simulazione in modo da ottenere 150 osservazioni della v.a. di interesse. I dati osservati sono stati raggruppati in gruppi di 10 osservazioni ciascuno, in modo da ottenere 15 campioni, ciascuno di numerosità n=10, ed è stata calcolata la media di ognuno. La media e la deviazione standard calcolate sui singoli campioni sono stime della media e della deviazione standard della popolazione. Con l'ausilio di MINITAB, sulla distribuzione delle medie campionarie è stato eseguito il test di Anderson-Darling per l'ipotesi di distribuzione normale delle medie campionarie, di cui si riportano i risultati:

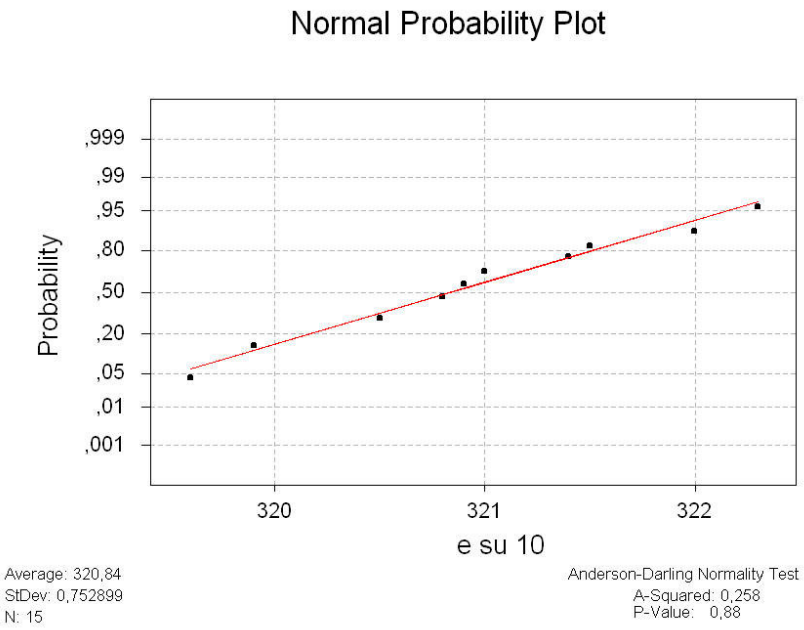



**Figura 6.16 Esempio di Piano di Produzione**

Il test di Anderson-Darling è una versione modificata del test di Kolmogorov-Smirnov, nel senso che fa uso della specifica distribuzione che si sta testando per il calcolo dei valori critici [55]. In Figura 6.17 si riportano i valori critici per la distribuzione Normale.

| $\alpha$ | 0.1 | 0.05 | 0.025 | 0.01 |
|---|---|---|---|---|
| CV | 0.631 | 0.752 | 0.873 | 1.035 |

**Figura 6.17 Valori critici della statistica AD nel caso di distribuzione normale**

Il test è a una coda e, fissato un livello di confidenza 1-$\alpha$, l'ipotesi che la distribuzione abbia la forma specificata viene rigettata se la statistica di Anderson-Darling

$AD^2 = -n - S$

$$S = \sum_{i=1}^{n} \frac{2i-1}{n} \{\ln F_o(z_i) + \ln(1 - F_0(z_{n+1-i}))\}$$

è maggiore del valore critico corrispondente. $F_0$ è la funzione di distribuzione cumulata della v.a. normale, $z_i$ è l'i-esimo valore standardizzato del campione mentre n è la dimensione campionaria [55].

Nel caso in esame AD=0.285, mentre il valore critico modificato in funzione della dimensione campionaria è, per $\alpha = 0.05$ e n=15

$$CV^* = \frac{0.752}{1 + \frac{0.75}{n} + \frac{2.25}{n}} = 0.709$$

Dato che risulta $AD < CV^*$ si accetta l'ipotesi di normalità delle medie campionarie calcolate su campioni di dimensione $n_c$=10. Il grafico riporta anche il p-value che rappresenta la probabilità di ottenere gli stessi risultati del test quando la normalità dei dati è reale.

A questo punto, trattandosi di una simulazione con terminazione, si è scelto di utilizzare la *Procedura Iterativa* per il calcolo del numero di repliche necessarie per ottenere una stima della media della v.a. *Montaggio_Traversa_Base_Imballo.NumberOut* con un margine di errore fissato. I motivi di questa scelta sono da ricercarsi nel fatto che nella procedura iterativa, ad ogni replica aggiuntiva eventualmente effettuata, viene ricalcolata la stima della varianza che, invece, nel caso della procedura a due fasi, rimane fissata a $S^2(n_0)$, fornendo quindi una stima più precisa della misura di prestazione di interesse. Per comodità, si riporta di seguito uno schema algoritmico della procedura.

È stato fissato un livello di confidenza $1 - \alpha = 0{,}95$ ed un errore relativo $\gamma = 0{,}10$ e si è posto $n_o = 10$ come dimensione campionaria iniziale. L'algoritmo è stato implementato con l'ausilio del MATLAB ed ha fornito $\frac{\delta(n,\alpha)}{\overline{X}(n)} \leq 0.09$ proprio per n=10. Dunque la stima del numero medio di congelatori impostato è pari a:

$$\overline{X}(10) = \frac{1}{10}\sum_{i=1}^{10} X_i = 320.84$$

mentre la semiampiezza del relativo intervallo di confidenza al 95% è:

$$\delta(10,0.05) = t_{10-1,0.975}\frac{S(10)}{\sqrt{10}} = 0.529$$

**Passo 1**: si effettuano $n_0$ repliche della simulazione e si pone $n = n_0$;

**Passo 2:** si calcolano $\overline{X}(n)$ e $\delta(n,\alpha)$ da $X_1,...,X_n$;

**Passo 3:** se $\frac{\delta(n,\alpha)}{\overline{X}(n)} \leq \frac{\gamma}{1+\gamma}$ si usa $\overline{X}(n)$ come stima di $\mu$ e STOP;
altrimenti si effettua un' ulteriore replica, si pone $n = n+1$ e si va al Passo 1.

## 6.4. Progettazione ed analisi degli esperimenti

Il modello di simulazione realizzato vuole essere un mezzo per valutare la fattibilità dei piani di produzione dell'azienda CO.PRO.
A titolo di esempio, si supponga che l'azienda abbia ricevuto, nella settimana corrente, gli ordini riportati in Figura 6.18:

**Ordini di Produzione**

| Giorni / Modelli | martedì | mercoledì | giovedì | venerdì |
|---|---|---|---|---|
| 1080 | | | 100 | |
| 10100 | 110 | | 80 | 50 |
| 10130 | 100 | | 80 | 50 |
| 10115 | | | 80 | |
| 10185 | 60 | | 100 | |
| 10270 | | | | 60 |
| 10300 | | 75 | | 60 |
| 10330 | | 50 | | |
| 10350 | | 100 | | 90 |
| 10400 | | 40 | | |
| 11330 | | | 30 | |
| 11410 | | | 30 | |

**Figura 6.18 Ordine di Produzione**

Per evadere gli ordini del martedì e del mercoledì parte della produzione è anticipata al lunedì secondo la seguente produzione (Figura 6.19):

| Giorni / Modelli | lunedì | Linea |
|---|---|---|
| 10100 | 110 | 1 |
| 10130 | 100 | 1 |
| 10185 | 60 | 2 |
| 10300 | 40 | 2 |
| 10350 | 50 | 3 |
| TOT | 360 | |

**Figura 6.19 Piano di Produzione**

Il mix produttivo, cioè il numero di maschere da dedicate a ciascuna linea, risulta direttamente proporzionale alla produzione programmata, perciò per la scelta del mix si può ricorrere alle seguenti proporzioni:

L1) $(TOT_{LINEA}/TOT)\cdot TOT_{SCHIUMATRICI}=(210/360)\cdot 6=3.50$

L2) $(TOT_{LINEA}/TOT)\cdot TOT_{SCHIUMATRICI}=(100/360)\cdot 6=1.67$ (e 6.1)

L3) $(TOT_{LINEA}/TOT)\cdot TOT_{SCHIUMATRICI}=(50/360)\cdot 6=0.83$

Dall'analisi delle L2) $(TOT_{LINEA}/TOT)\cdot TOT_{SCHIUMATRICI}=(100/360)\cdot 6=1.67$ (e 6.1) risulta che l'azienda può scegliere tra due possibili mix produttivi: il (3,2,1), con la prima linea sovrasatura e le linee 2 e 3 insature, oppure il mix (4,1,1), con la prima e la terza linea insature e la 2 sovrasatura. Nel primo caso l'azienda effettua un set-up lungo che sposta la schiumatrice dedicata alla terza linea sulla prima, mentre nel secondo caso realizza un set-up lungo che sposta la schiumatrice della terza linea sulla seconda. In ogni caso l'azienda opterà per un set-up lungo di un'ora che, dunque, non viene considerato nella successiva analisi dei fattori. Lo stesso avviene per il set-up breve di 15 minuti che si rende necessario sulle linee 1 e 2 per il passaggio della schiumatrice dal primo al secondo modello da realizzare. Poiché la produzione prevista supera di gran lunga la stima del numero medio di congelatori impostato, può rendersi necessario

effettuare uno straordinario. Per selezionare la politica migliore, si ricorre alla progettazione di un piano degli esperimenti.
Sono stati individuati i seguenti fattori di interesse di cui si vogliono stimare gli effetti sul numero di congelatori impostato e sull'utilizzazione delle schiumatici (Figura 6.20):

| **FATTORI** | **LIVELLO** | |
|---|---|---|
| | **BASSO** | **ALTO** |
| **Mix produttivo** | (3,2,1) | (4,1,1) |
| **Straordinario** | Assente=8 ore lavorative | Presente=9 ore lavorative |

**Figura 6.20 Fattori di interesse**

Riassumendo, mediante la progettazione ed analisi degli esperimenti si vogliono individuare i fattori che hanno la più alta influenza sulle prestazioni dell'azienda e sui quali l'azienda può intervenire per migliorare la sua produttività e per aumentare l'utilizzazione delle risorse. Il fattore Mix è di tipo qualitativo, mentre il fattore Straordinario è di tipo quantitativo. Entrambi sono caratterizzati da due soli livelli.
Per la valutazione degli effetti dei fattori di interesse e della loro interazione, si è scelto di simulare un *piano fattoriale completo* $2^2$ *completamente casualizzato,* ovvero di simulare le combinazioni dei livelli dei fattori riportate in Figura 6.21:

| *Trattamenti* | *Mix* | *Straordinario* |
|---|---|---|
| $T_1$ | (3,2,1) | 8 |
| $T_2$ | (4,1,1) | 9 |
| $T_3$ | (3,2,1) | 9 |
| $T_4$ | (4,1,1) | 8 |

**Figura 6.21 Piano degli esperimenti**

Avendo scelto un piano fattoriale completamente casualizzato, l'ordine secondo cui verranno implementati i trattamenti, ovvero l'ordine con cui saranno eseguite le simulazioni dei diversi scenari, non influenzerà in alcun modo la stima degli effetti dei fattori di interesse.
Nella teoria degli esperimenti una decisione critica è rappresentata dalla scelta del numero di repliche di ogni trattamento. Uno dei metodi più usati per

determinare il numero di repliche è quello che utilizza le *curve operative caratteristiche*. Queste rappresentano l'andamento della probabilità dell'errore di IIª specie $\beta$ in funzione di un parametro $\phi$ che dipende dal numero di repliche *n*, dai gradi di libertà della *F di Fischer* e dal rischio di errore di Iª specie $\alpha$ [120]. L'errore di IIª specie $\beta_i$ è per definizione la probabilità di accettare l'ipotesi nulla $H_o$ quando è falsa, essendo vera l'ipotesi alternativa $H_i$. Il complemento all'unità dell'errore di IIª specie, $1-\beta_i$, è la probabilità di rigettare $H_o$ quando è falsa ed esprime il *potere* del test nel diagnosticare quanto previsto dalla corrispondente $H_i$ [62].

Nella Figura 6.22 si riportano le formule per il calcolo di $\phi$ per un progetto fattoriale a due fattori.

| Fattori | $\phi^2$ | Gradi di libertà del numeratore | Gradi di libertà del denominatore |
|---|---|---|---|
| A | $\frac{bn\sum_{i=1}^{a}\tau_i^2}{a\sigma^2}$ | a-1 | ab(n-1) |
| B | $\frac{an\sum_{i=1}^{b}\beta_i^2}{b\sigma^2}$ | b-1 | ab(n-1) |
| AB | $\frac{n\sum_{i=1}^{a}\sum_{j=1}^{b}(\tau\beta)_{ij}^2}{\sigma^2\left[(a-1)(b-1)+1\right]}$ | (a-1)(b-1) | ab(n-1) |

**Figura 6.22 Formule per il calcolo di $\phi^2$**

Il calcolo del parametro $\phi$ richiede una stima degli effetti dei fattori e della varianza $\sigma^2$ dell'errore. Per poter stimare tali parametri sono eseguite tre repliche dell'esperimento. Nella Figura 6.23 si riporta il numero medio di congelatori impostato ottenuto ad ogni replica e in corrispondenza di ciascun trattamento.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Mix** | **Straordinario** | | | | | | | |
| | | **Livello 1** | | | **Livello 2** | | | |
| | **Livello 1** | 320.6 | 321 | 320.3 | 357.8 | 357.5 | 357.5 | 2034.7 |
| | **Livello 2** | 308.9 | 308.7 | 308.9 | 325 | 325.3 | 324.6 | 1901.4 |

| | | | | |
|---|---|---|---|---|
| | | 1888.4 | 2047.7 | 3936.1 |

**Figura 6.23 Risultati delle prime tre repliche dei trattamenti**

Le stime della media globale e degli effetti dei trattamenti sono date da:

$$\hat{\mu} = \bar{y}...$$
$$\hat{\tau}_i = \bar{y}_{i}.. - \bar{y}...$$
$$\hat{\beta}_i = \bar{y}_{.j}. - \bar{y}...$$
$$(\tau\beta)_{ij} = \bar{y}_{ij}. - \bar{y}_{i}.. - \bar{y}_{.j}. + \bar{y}...$$

dove:

$$\bar{y}... = \frac{1}{2 \cdot 2 \cdot 3}\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{3} y_{ijk}$$
$$\bar{y}_{i}.. = \frac{1}{2 \cdot 3}\sum_{j=1}^{2}\sum_{k=1}^{3} y_{ijk}$$
$$\bar{y}_{.j}. = \frac{1}{2 \cdot 3}\sum_{i=1}^{2}\sum_{k=1}^{3} y_{ijk}$$
$$\bar{y}_{ij}. = \frac{1}{3}\sum_{k=1}^{3} y_{ijk}$$

Effettuando i calcoli si ottengono le seguenti stime:

$$\hat{\mu} = 328$$
$$\hat{\tau}_1 = 5$$
$$\hat{\tau}_2 = -4.9$$
$$\hat{\beta}_1 = -20.77$$
$$\hat{\beta}_2 = 20.85$$
$$(\tau\beta)_{11} = 1.6$$
$$(\tau\beta)_{12} = -1.45$$
$$(\tau\beta)_{21} = -1.2$$
$$(\tau\beta)_{22} = 1.45$$

La stima della varianza $\sigma^2$ è fornita dal valore atteso dello scarto quadratico medio dell'errore:

$$\sigma^2 = E(MS_E) = E\left(\frac{SS_E}{ab(n-1)}\right) = 0.16 .$$

A partire da queste stime è stato calcolato il parametro $\phi^2$:

| Fattori | $\phi^2$ | Gradi di libertà del numeratore | Gradi di libertà del denominatore |
|---|---|---|---|
| Mix | 31.3 | 1 | 8 |
| Straordinario | 29.4 | 1 | 8 |
| Interazione | 2.6 | 1 | 8 |

**Figura 6.24 Calcolo** $\phi^2$

Per il calcolo di n si è scelto il più piccolo valore di $\phi^2$ ottenuto dalle tre repliche dell'esperimento, perché, fissato n, in corrispondenza del minimo valore di $\phi^2$ si ottiene il più piccolo valore della potenza del test inferiore a quello specificato nell'esperimento; in questo modo, incrementando n, $\phi^2_{\min}$ aumenta e con esso anche il potere del test [120].

Posto $\alpha = 0.05$ e $1-\beta = 0.90$, entrando nel diagramma in Figura 6.25 con $\phi_{\min} = 1.61$, $\nu_1 = 1$ e $\nu_2 = 8$, si legge $\beta = 0.40$:

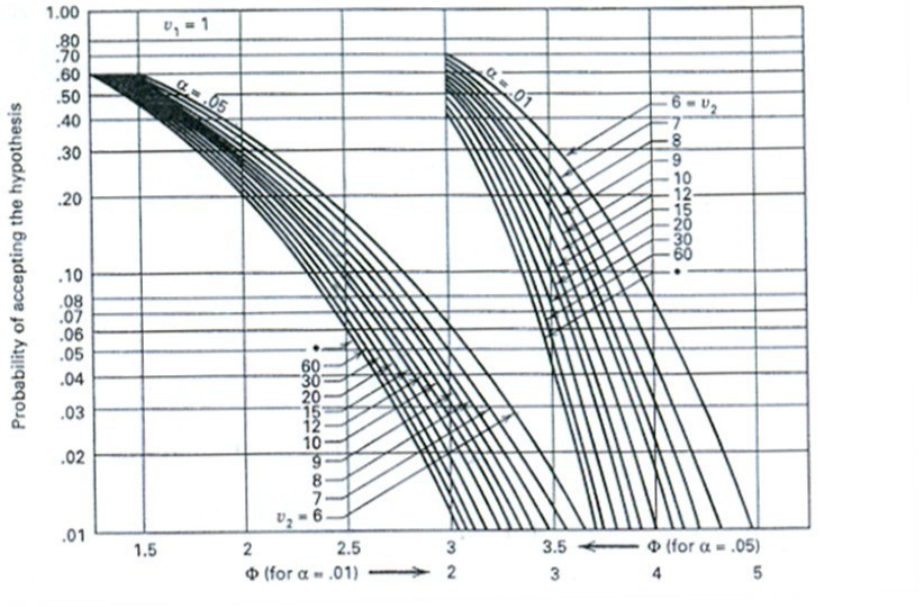



**Figura 6.25 Curve operative caratteristiche per il calcolo di n**

Risulta evidente che tre repliche dell'esperimento non sono sufficienti. A questo punto si procede per tentativi, incrementando n di un'unità fino a che non risulta $\beta \le 0.10$ (Figura 6.26):

| n | $\phi^2$ | $\phi$ | $\nu_1$ | $\nu_2$ | $\beta$ |
|---|---|---|---|---|---|
| 4 | **$0.86 \cdot 4 = 3.48$** | 1.86 | 1 | 12 | 0.36 |
| 5 | **$0.86 \cdot 4 = 3.48$** | 2.09 | 1 | 16 | 0.18 |
| 6 | **$0.86 \cdot 4 = 3.48$** | 2.28 | 1 | 20 | 0.13 |

| | | | | | |
|---|---|---|---|---|---|
| 7 | **$0.86 \cdot 4 = 3.48$** | 2.47 | 1 | 24 | 0.082 |

**Figura 6.26 Risultati dei tentativi per il calcolo di n**

La potenza del test risulta superiore a quello specificato per n=7. Si è scelto, dunque, di eseguire sette repliche per ogni trattamento.

Le repliche sono ciascuna composta da 10 ripetizioni della simulazione di ogni trattamento, in modo da avere una stima precisa del numero medio di congelatori impostati in ogni scenario. Con l'ausilio del MINITAB è stato creato un piano fattoriale completo $2^2$ completamente casualizzato; in questo modo l'ordine con cui vengono implementati i trattamenti, ovvero l'ordine con il quale si effettuano le simulazioni, non incide sulla stima degli effetti dei fattori. Il MINITAB ha fornito le combinazioni di Figura 6.27.

| RunOrder | Mix | Straordinario | Impostato |
|---|---|---|---|
| 1 | (3,2,1) | 9 | 357.8 |
| 2 | (3,2,1) | 8 | 320.6 |
| 3 | (4,1,1) | 8 | 308.9 |
| 4 | (4,1,1) | 9 | 325 |
| 5 | (3,2,1) | 8 | 321 |
| 6 | (3,2,1) | 9 | 357.5 |
| 7 | (3,2,1) | 8 | 320.3 |
| 8 | (4,1,1) | 9 | 325.3 |
| 9 | (4,1,1) | 8 | 308.7 |
| 10 | (3,2,1) | 8 | 320.8 |
| 11 | (4,1,1) | 9 | 324.6 |
| 12 | (4,1,1) | 8 | 308.9 |
| 13 | (4,1,1) | 8 | 309 |
| 14 | (4,1,1) | 9 | 325.4 |
| 15 | (4,1,1) | 8 | 308.4 |
| 16 | (3,2,1) | 9 | 357.5 |
| 17 | (3,2,1) | 9 | 357.2 |
| 18 | (3,2,1) | 9 | 357.4 |
| 19 | (3,2,1) | 9 | 357.2 |
| 20 | (4,1,1) | 9 | 324.8 |
| 21 | (4,1,1) | 9 | 324.9 |
| 22 | (4,1,1) | 9 | 325.2 |
| 23 | (3,2,1) | 8 | 320.3 |
| 24 | (4,1,1) | 8 | 308.7 |
| 25 | (4,1,1) | 8 | 309.2 |
| 26 | (3,2,1) | 9 | 357.2 |
| 27 | (3,2,1) | 8 | 321.2 |
| 28 | (3,2,1) | 8 | 321 |

**Figura 6.27 Worksheet del MINITAB**

La colonna "Impostato" contiene le stime del numero di congelatori impostato in ogni trattamento e su questi dati è stata applicata l'analisi della varianza a due vie.

Nel caso di piano fattoriale $2^2$ completo, il modello degli effetti è il seguente:

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} \qquad \begin{matrix} i = 1,2 \\ j = 1,2 \\ k = 1,...7 \end{matrix}$$

dove:

- $\mu$ è l'effetto medio globale, $E(y_{ijk}) = \mu$, ovvero è il risultato che si avrebbe se tutti i trattamenti fossero uguali;
- $\tau_i$ è l'effetto dovuto al livello i-esimo del fattore A (Mix produttivo);
- $\beta_j$ è l'effetto del livello j-esimo del fattore B (Straordinario);
- $(\tau\beta)_{ij}$ è l'effetto dovuto all'interazione tra i due fattori:
- $\varepsilon_{ijk}$ è la variabile aleatoria "errore sperimentale".

In quanto errori sperimentali, è del tutto plausibile supporre che le v.a. $\varepsilon_{ijk}$ siano distribuite secondo Gaussiane s-indipendenti con media nulla e varianza $\sigma^2$ costante per tutti i livelli dei fattori. Questa ipotesi implica che le osservazioni $y_{ijk}$ siano anch'esse distribuite secondo Gaussiane s-indipendenti con media $\mu + \tau_i + \beta_j + (\tau\beta)_{ij}$ e varianza $\sigma^2$:

$$y_{ijk} \approx N(\mu + \tau_i + \beta_j + (\tau\beta)_{ij}; \sigma^2)$$

Avendo posto $E(y_{ijk}) = \mu_{ij} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij}$, allora deve necessariamente essere

$$\sum_{i=1}^{a} \tau_i = 0 \qquad \sum_{j=1}^{b} \beta_j = 0 \qquad \sum_{i=1}^{a} (\tau\beta)_{ij} = \sum_{j=1}^{b} (\tau\beta)_{ij} = 0$$

cioè gli effetti di un fattore, quando sono significativi, possono essere considerati come deviazione dalla media globale. Dunque, se gli effetti dei fattori Mix e Straordinario e della loro interazione non sono significativi, ovvero non influenzano il numero di congelatori impostato, applicando un test di ipotesi risulteranno vere le seguenti ipotesi nulle:

$$H_0 : \tau_1 = \tau_2 = 0$$

$$H_0 : \beta_1 = \beta_2 = 0$$

$$H_0 : (\tau\beta)_{ij} = 0$$

mentre se gli effetti sono significativi risulteranno vere le seguenti ipotesi alternative:

$$H_1 : almeno\ un\ \tau_i \neq 0$$

$$H_1 : almeno\ un\ \beta_i \neq 0$$

$$H_1 : almeno\ un\ (\tau\beta)_{ij} \neq 0$$

La procedura utilizzata per testare l'ipotesi di uguaglianza delle medie dei trattamenti è l'analisi della varianza a due vie, rappresentata nella Figura 6.28 per il caso in esame.

| Fonti di variabilità | Somma dei quadrati degli scarti | Gradi di libertà | Scarto quadratico medio | $F_0$ |
|---|---|---|---|---|
| Mix produttivo | $SS_{Mix}$ | 2-1 | $MS_{Mix} = \frac{SS_{Mix}}{2-1}$ | $F_0 = \frac{MS_{Mix}}{MS_{Err}}$ |
| Straordinario | $SS_{Straordinario}$ | 2-1 | $MS_{Straor} = \frac{SS_{Straor}}{2-1}$ | $F_0 = \frac{MS_{Straor}}{MS_{Err}}$ |
| Interazione | $SS_{interazione}$ | (2-1)(2-1) | $MS_{Inter} = \frac{SS_{Inter}}{(2-1)(2-1)}$ | $F_0 = \frac{MS_{Inter}}{MS_{Err}}$ |
| Errore | $SS_{Errore}$ | $2 \cdot 2 \cdot (7-1)$ | $MS_{Err} = \frac{SS_{Err}}{2 \cdot 2 \cdot (7-1)}$ | |
| Totale | $SS_T$ | $2 \cdot 2 \cdot 7 - 1$ | | |

**Figura 6.28 Analisi della varianza a due vie**

L'analisi della varianza consente di ripartire la varianza sperimentale in aliquote s-indipendenti al fine di isolare quelle imputabili ai fattori per il cui studio è stato formulato il piano sperimentale. Per isolare l'aliquota, della varianza globale, imputabile ai fattori in esame e alla loro interazione si consideri la somma totale dei quadrati degli scarti:

$$SS_T = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{7} y_{ijk}^2 - \frac{y^2\ldots}{28}$$

Essa rappresenta una misura della variabilità dei dati, che divisa per i gradi di libertà totali, abn-1, costituisce la varianza campionaria delle osservazioni. Tale somma può essere ripartita in quattro aliquote s-indipendenti:

$$SS_T = SS_{Mix} + SS_{Straor} + SS_{Inter} + SS_{Err}$$

dove:

$SS_{Mix}$ è la somma dei quadrati delle differenze tra la media delle osservazioni dell'i-esimo trattamento del fattore Mix e la media globale:

$$SS_{Mix} = 14\sum_{i=1}^{2}\left(\bar{y}_{i\cdot\cdot} - \bar{y}_{\cdot\cdot\cdot}\right)^2 ;$$

$SS_{Straor}$ è la somma dei quadrati delle differenze tra la media delle osservazioni del j-esimo trattamento del fattore Straordinario e la media globale:

$$SS_{Straor} = 14\sum_{j=1}^{2}\left(\bar{y}_{\cdot j\cdot} - \bar{y}_{\cdot\cdot\cdot}\right)^2 ;$$

$SS_{Inter}$ è la somma dei quadrati dovuta all'interazione tra i due fattori in esame:

$$SS_{Inter} = 7\sum_{i=1}^{2}\sum_{j=1}^{2}\left(\bar{y}_{ij\cdot} - \bar{y}_{i\cdot\cdot} - \bar{y}_{\cdot j\cdot} + \bar{y}_{\cdot\cdot\cdot}\right)^2 ;$$

$SS_{Err}$ è la somma dei quadrati delle differenze tra le singole osservazioni dell'i-esimo trattamento e la media delle osservazioni del medesimo:

$$SS_{Err} = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{7}\left(y_{ijk} - \bar{y}_{ij\cdot}\right)^2 .$$

Se si divide ciascun termine della somma per i rispettivi gradi di libertà, si ottengono gli scarti quadratici medi, i cui valori attesi sono:

$$E(MS_{Mix}) = E\left(\frac{SS_{Mix}}{2-1}\right) = \sigma^2 + \frac{14\sum_{i=1}^{2}\tau_i^2}{2-1}$$

$$E(MS_{Straor}) = E\left(\frac{SS_{Straor}}{2-1}\right) = \sigma^2 + \frac{14\sum_{j=1}^{2}\beta_j^2}{2-1}$$

$$E(MS_{Inter}) = E\left(\frac{SS_{Inter}}{(2-1)(2-1)}\right) = \sigma^2 + \frac{7\sum_{i=1}^{2}\sum_{j=1}^{2}(\tau\beta)_{ij}^2}{(2-1)(2-1)}$$

$$E(MS_{Err}) = E\left(\frac{SS_{Err}}{2\cdot 2\cdot(7-1)}\right) = \sigma^2 .$$

Se è vera l'ipotesi nulla di effetti non significativi, gli scarti quadratici medi sono tutte stime della varianza globale $\sigma^2$; se, invece, ci sono differenze tra le medie dei trattamenti è vera l'ipotesi alternativa e gli scarti quadratici medi $MS_{Mix}$, $MS_{Straor}$, $MS_{Inter}$ sono tutti valori più grandi di $MS_{Err}$. Inoltre, se è vera l'ipotesi di errore normalmente e indipendentemente distribuito con varianza costante, i rapporti

$$\frac{MS_{Mix}}{MS_{Err}} \quad \frac{MS_{Straor}}{MS_{Err}} \quad \frac{MS_{Inter}}{MS_{Err}}$$

si distribuiscono come variabili aleatorie di Fisher, rispettivamente con $2-1$, 2-1, (2-1)(2-1) gradi di libertà al numeratore e $2\cdot 2\cdot(7-1)$ gradi di libertà al denominatore, essendo rapporti tra due v.a. *Chi-Quadrato*, divise per i rispettivi gradi di libertà. Detto questo, si rigetta l'ipotesi $H_0$ di uguaglianza delle medie dei trattamenti di ciascun fattore con un livello di significatività $1-\alpha$ se il valore del rapporto

$$F_0 = \frac{MS_{fattore}}{MS_{errore}}$$

è maggiore del percentile $F_{\alpha,DFnumeratore,DFdenominatore}$ della distribuzione di Fisher.

L'analisi della varianza può essere applicata solo nel caso in cui l'ipotesi di s-indipendenza e di distribuzione normale della v.a. "errore sperimentale" sia verificata. La validità di quest'ipotesi può essere testata mediante un'analisi grafica dei residui, che per un progetto fattoriale $2^2$ sono definiti come:

$$e_{ijk} = y_{ijk} - \hat{y}_{ijk}$$

con $\hat{y}_{ijk} = \bar{y}_{ij}.$. Se l'ipotesi di normalità è valida, il grafico di probabilità normale dei residui (Figura 6.29) assomiglia ad una linea retta. Il valore dei residui per il caso in esame è riportato in Figura 6.30.

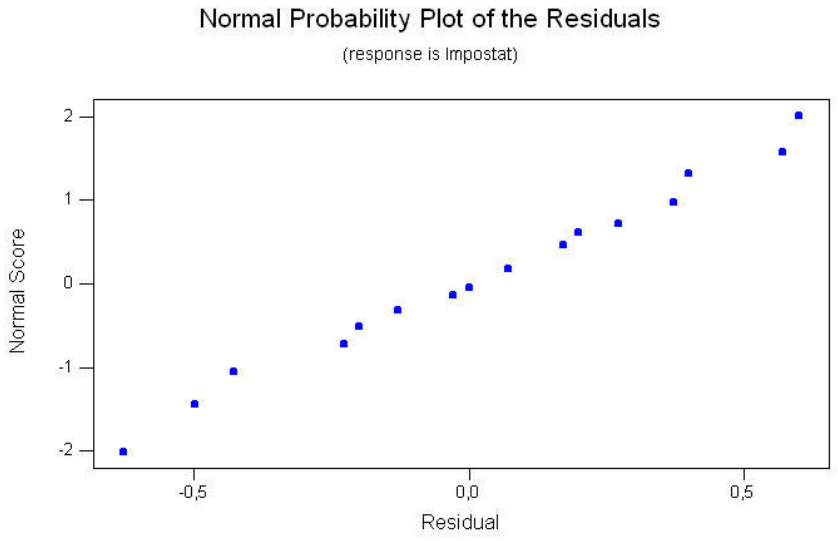



**Figura 6.29 Grafico di probabilità normale dei residui**

Observations for Impostat

| Obs | Impostat | Fit | SE Fit | Residual | St Resid |
|---|---|---|---|---|---|
| 1 | 357,800 | 357,400 | 0,108 | 0,400 | 1,52 |
| 2 | 320,600 | 320,743 | 0,108 | -0,143 | -0,54 |
| 3 | 308,900 | 308,829 | 0,108 | 0,071 | 0,27 |
| 4 | 325,000 | 325,029 | 0,108 | -0,029 | -0,11 |
| 5 | 321,000 | 320,743 | 0,108 | 0,257 | 0,98 |
| 6 | 357,500 | 357,400 | 0,108 | 0,100 | 0,38 |
| 7 | 320,300 | 320,743 | 0,108 | -0,443 | -1,68 |
| 8 | 325,300 | 325,029 | 0,108 | 0,271 | 1,03 |
| 9 | 308,700 | 308,829 | 0,108 | -0,129 | -0,49 |
| 10 | 320,800 | 320,743 | 0,108 | 0,057 | 0,22 |
| 11 | 324,600 | 325,029 | 0,108 | -0,429 | -1,63 |
| 12 | 308,900 | 308,829 | 0,108 | 0,071 | 0,27 |
| 13 | 309,000 | 308,829 | 0,108 | 0,171 | 0,65 |
| 14 | 325,400 | 325,029 | 0,108 | 0,371 | 1,41 |
| 15 | 308,400 | 308,829 | 0,108 | -0,429 | -1,63 |
| 16 | 357,500 | 357,400 | 0,108 | 0,100 | 0,38 |
| 17 | 357,200 | 357,400 | 0,108 | -0,200 | -0,76 |
| 18 | 357,400 | 357,400 | 0,108 | 0,000 | 0,00 |
| 19 | 357,200 | 357,400 | 0,108 | -0,200 | -0,76 |
| 20 | 324,800 | 325,029 | 0,108 | -0,229 | -0,87 |
| 21 | 324,900 | 325,029 | 0,108 | -0,129 | -0,49 |
| 22 | 325,200 | 325,029 | 0,108 | 0,171 | 0,65 |
| 23 | 320,300 | 320,743 | 0,108 | -0,443 | -1,68 |
| 24 | 308,700 | 308,829 | 0,108 | -0,129 | -0,49 |
| 25 | 309,200 | 308,829 | 0,108 | 0,371 | 1,41 |
| 26 | 357,200 | 357,400 | 0,108 | -0,200 | -0,76 |
| 27 | 321,200 | 320,743 | 0,108 | 0,457 | 1,73 |
| 28 | 321,000 | 320,743 | 0,108 | 0,257 | 0,98 |

Estimated Coefficients for Impostat using data in uncoded units

| Term | Coef |
|---|---|
| Constant | 328,000 |
| Mix | -11,0714 |
| Straordi | 13,2143 |
| Mix*Straordi | -5,11429 |

**Figura 6.30 Analisi dei residui**

Nello specifico, il normal plot dei residui non mostra anomalie evidenti: i dati si distribuiscono seguendo una linea retta e non si evidenzia la presenza di outliers. Tuttavia, l'analisi può essere approfondita per svincolarla dalla soggettività dell'osservatore. Per verificare la presenza o meno di outliers si analizzano i residui standardizzati, riportati in Figura 6.30:

$$d_{ijk} = \frac{e_{ijk}}{\sqrt{MS_{Err}}}.$$

Se l'errore è una v.a. normale con media nulla e varianza $\sigma^2$, i residui standardizzati hanno una distribuzione approssimativamente normale con media nulla e varianza unitaria.

L'ipotesi di indipendenza degli errori può essere verificata analizzando il grafico dei residui in funzione dell'ordine dei dati: se esso non mostra un andamento specifico allora l'ipotesi è valida. Il grafico dei residui in funzione dell'ordine dei dati è riportato in Figura 6.31.

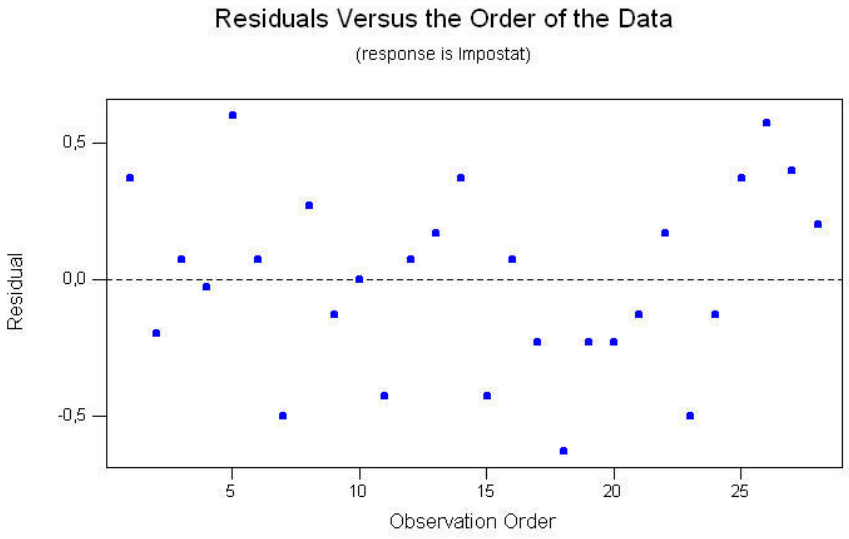



**Figura 6.31 Grafico dei residui in funzione dell'ordine dei dati**

In prima analisi non ci sono ragioni per sospettare alcuna violazione dell'ipotesi di indipendenza stocastica degli errori, come era da aspettarsi avendo scelto un piano sperimentale completamente casualizzato.

L'ipotesi di varianza costante viene verificata analizzando il grafico dei residui in funzione dei valori attesi delle risposte, riportati in Figura 6.32.

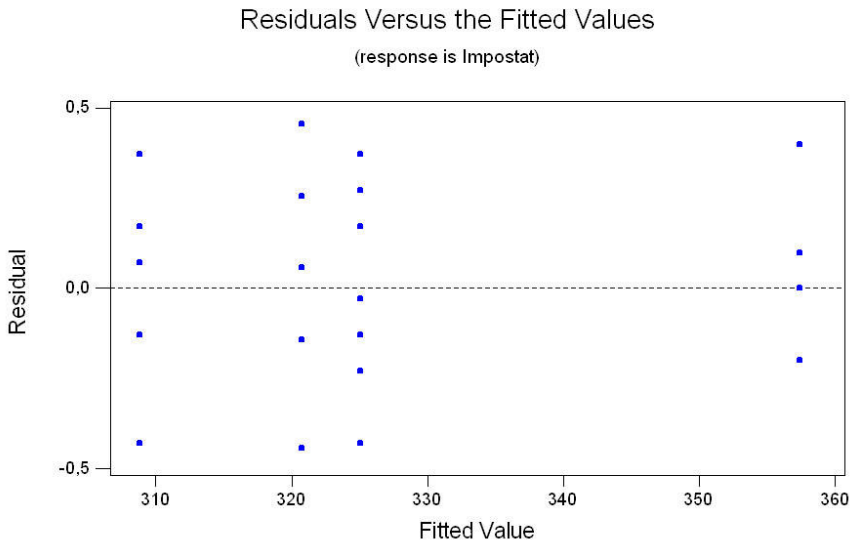



**Figura 6.32 Grafico dei residui in funzione delle stime delle risposte**

Anche in questo caso il grafico non mostra un andamento particolare, sicchè è possibile ritenere valida anche l'ipotesi di varianza costante.
Verificata l'adeguatezza del modello, è possibile passare all'analisi della varianza. Nella finestra *Session* del MINITAB (Figura 6.33) leggiamo:

```
Analysis of Variance for Impostat

Source         DF        SS        MS        F        P
Mix             1  3432,143  3432,143  4,2E+04    0,000
Straordi        1  4889,286  4889,286  6,0E+04    0,000
Interaction     1   732,366   732,366  9033,59    0,000
Error          24     1,946     0,081
Total          27  9055,740
```

**Figura 6.33 Session del MINITAB**

Avendo scelto un livello di significatività per il test 1-$\alpha$=0.95, il percentile rispetto al quale confrontare le statistiche $F_0$ è pari a :

$$F_{0.05,1,24} = 4.26$$

Poiché risulta:

$$F_0 = 4.2 \cdot 10^4 > 4.26$$
$$F_0 = 6.0 \cdot 10^4 > 4.26$$
$$F_0 = 9033.59 > 4.26$$

si rigetta l'ipotesi $H_0$ e si conclude che le medie dei trattamenti differiscono, cioè sia lo straordinario, sia la scelta del mix produttivo influenzano il numero di congelatori impostato. Nella Figura 6.34 si riportano le stime degli effetti fornite dal MINITAB:

Estimated Effects and Coefficients for Impostat (coded units)

| Term | Effect | Coef | SE Coef |
|---|---|---|---|
| Constant | | 328,00 | 0,05381 |
| Mix | -22,14 | -11,07 | 0,05381 |
| Straordi | 26,43 | 13,21 | 0,05381 |
| Mix*Straordi | -10,23 | -5,11 | 0,05381 |

**Figura 6.34 Stima degli effetti**

Analizzando il grafico delle interazioni di Figura 6.35, si può notare che, sia in assenza di straordinario, sia in presenza di un'ora di straordinario, il mix produttivo (3,2,1) ha un effetto maggiore sul numero di congelatori impostati rispetto al mix (4,1,1); quindi, qualunque sia il livello del fattore straordinario (assente o presente), il mix (3,2,1) risulta migliore in termini di produttività rispetto al mix (4,1,1), come si evince anche dal grafico degli effetti principali di Figura 6.36.

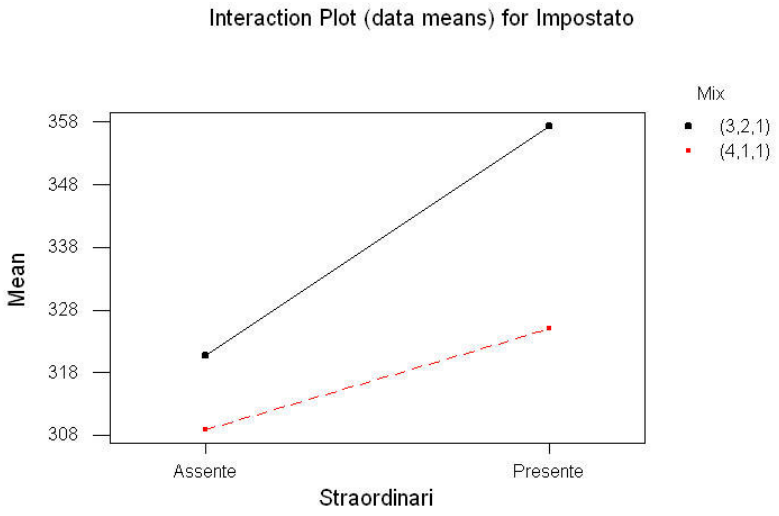



**Figura 6.35 Grafico delle interazioni**

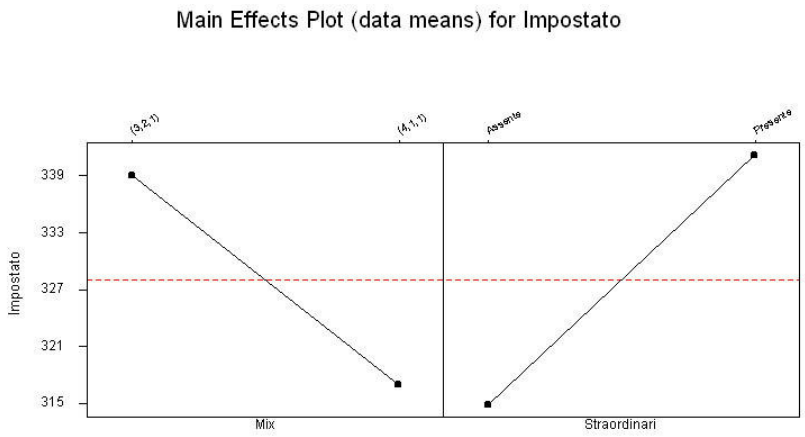



**Figura 6.36 Grafico degli effetti principali**

Dalla Figura 6.35 e dalla Figura 6.36 risulta inoltre evidente come la presenza dello straordinario incrementi il numero di congelatori impostato.

In Figura 6.37 si riportano le stime del numero medio di congelatori impostato con ciascun mix produttivo e i relativi intervalli di confidenza al 95%. Gli intervalli di confidenza per le medie dei trattamenti si ottengono utilizzando la formula:

$$\mu \pm t_{n-1,1-\frac{\alpha}{2}} \sqrt{\frac{MS_E}{n}}$$

mentre gli intervalli di confidenza per le differenze tra le medie di due trattamenti si ricavano con la formula:

$$\Delta\mu \pm q_\alpha(a,f)\sqrt{\frac{MS_E}{n}}$$

con n=7, $a$=2, $f$=24.

| ***Straordinario*** | ***Mix (3,2,1)*** | ***Mix (4,1,1)*** | ***Differenza tra le medie*** |
|---|---|---|---|
| ***8*** | **320.74 ± 0.26** | **308.83 ± 0.26** | **11.91 ± 0.31** |
| ***9*** | **357.4 ± 0.26** | **325.03 ± 0.26** | **32.37 ± 0.31** |
| ***Differenza tra le medie*** | **36.66 ± 0.31** | **16.20 ± 0.31** | |

**Figura 6.37 Intervalli di confidenza per le medie dei trattamenti**

Le cause che hanno portato a ritenere migliore, in termini di produttività, il mix (3,2,1) possono essere indagate analizzando l'utilizzazione delle schiumatrici. Tutte le repliche delle simulazioni dei diversi scenari del modello hanno sempre fornito lo stesso valore di questi parametri, dunque si è fatta l'ipotesi che l'utilizzazione delle schiumatrici possa essere considerato un parametro costante. Nella Figura 6.38 si riportano i valori dell'utilizzazione delle schiumatrici forniti dalle simulazioni:

| ***Schiumatrice*** | ***Mix (3,2,1)*** | ***Mix (4,1,1)*** |
|---|---|---|
| 1 | 0.91 | 0.8825 |
| 2 | 0.89 | 0.8537 |
| 3 | 0.77 | 0.6986 |
| 4 | **0.85** | **0.1839** |
| 5 | 0.85 | 0.975 |
| 6 | 0.94 | 0.94 |

**Figura 6.38 Utilizzazione delle schiumatrici**

Risulta evidente una sottoutilizzazione della quarta schiumatrice nel caso in cui l'azienda decida di adottare il mix produttivo (4,1,1), a causa di un basso tasso di arrivo al forno di preriscaldamento della linea 1.

**Costruzione del metamodello**

Il modello di simulazione non è che una funzione non nota e nella maggior parte dei casi piuttosto complicata. Tuttavia, è possibile sviluppare formule matematiche che approssimino questa funzione e che siano in grado di indicare cosa succede agli output al variare delle combinazioni dei parametri di input. Le formule non sono altro che modelli matematici del modello di simulazione, definiti *metamodelli* o anche *modelli di regressione*. Conoscere il metamodello che meglio approssima il modello di simulazione può essere utile per prevedere le risposte della simulazione quando questa è lunga e costosa. In generale, se c'è una singola variabile di risposta che dipende da k variabili indipendenti, dette regressori, la relazione tra queste variabili è del tipo:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \varepsilon \qquad \text{(e 6.2)}$$

definito *modello di regressione lineare multiplo,* dove:

$\beta_j$ sono i coefficienti di regressione, non noti ma stimabili;

$\varepsilon$ è l'errore casuale, o rumore, che rappresenta l'inaccuratezza del modello matematico nell'approssimare la risposta *y* del modello di simulazione.

Nel caso che si sta analizzando, la variabile *numero di congelatori impostato* dipende dai due fattori "mix produttivo" e "straordinario", ciascuno caratterizzato da due livelli, e dalla loro interazione. Il modello di regressione che meglio esprime la relazione tra le variabili in studio è il seguente:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \varepsilon \qquad \text{(e 6.3)}$$

dove $\beta_{12} x_1 x_2$ esprime la dipendenza di *y* dall'interazione dei due fattori.

Se si pone $x_1 x_2 = x_3$ e $\beta_{12} = \beta_3$, l'equazione $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \varepsilon$ (e 6.2) è del tutto simile alla $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \varepsilon$ (e 6.3).

I regressori possono essere inseriti all'interno del modello sia come variabili codificate, sia come livelli naturali dei fattori che rappresentano. Nel caso in esame, il regressore $x_S$ relativo allo straordinario assume valore +1 se lo straordinario è pari ad un'ora e valore -1 se esso è assente (turno lavorativo di 8 ore). Analogamente, il regressore $x_M$ relativo al mix produttivo assume valore +1 in corrispondenza del mix (3,2,1) e valore -1 per il mix (4,1,1). Se i regressori sono rappresentati da variabili codificate, è possibile dimostrare, mediante il metodo dei minimi quadrati, che l'intercetta $\beta_0$ è la media globale di tutte le osservazioni, mentre i coefficienti di regressione $\beta_j, j = 1,2,...,k$, del modello sono la metà degli effetti principali dei fattori a cui sono associati:

$$y = \mu + \left(\frac{EFFETTO_A}{2}\right) x_A + \left(\frac{EFFETTO_B}{2}\right) x_B + ... + \left(\frac{EFFETTO_K}{2}\right) x_K$$

Invece, se i regressori sono espressi come livelli naturali dei fattori, i coefficienti si ottengono mediante le seguenti, semplici trasformazioni:

$$x_A = \frac{A - (A^- + A^+)/2}{(A^+ - A^-)/2}$$
$$x_B = \frac{B - (B^- + B^+)/2}{(B^+ - B^-)/2} \qquad \text{(e 6.4)}$$
$$\vdots$$
$$x_K = \frac{K - (K^- + K^+)/2}{(K^+ - K^-)/2}$$

Il modello di regressione diventa:

$$y = \mu + \left(\frac{EFFETTO_A}{2}\right)\left(\frac{A - (A^- + A^+)/2}{(A^+ - A^-)/2}\right) + \left(\frac{EFFETTO_B}{2}\right)\left(\frac{B - (B^- + B^+)/2}{(A^+ - A^-)/2}\right) +$$
$$+ \ldots + \left(\frac{EFFETTO_K}{2}\right)\left(\frac{K - (K^- + K^+)/2}{(K^+ - K^-)/2}\right)$$

Dalle relazioni $x_A = \frac{A - (A^- + A^+)/2}{(A^+ - A^-)/2}$, $x_B = \frac{B - (B^- + B^+)/2}{(B^+ - B^-)/2}$, $\vdots$, $x_K = \frac{K - (K^- + K^+)/2}{(K^+ - K^-)/2}$ (e 6.4)è facile intuire che se uno dei fattori è di tipo qualitativo verrà espresso sempre mediante variabili codificate.

Per il progetto fattoriale in studio, il MINITAB fornisce le seguenti stime dei coefficienti del modello di regressione Figura 6.39:

Stima dei coefficienti con i regressori espressi mediante variabili non codificate

| Term | Coef |
|---|---|
| Constant | 103,357 |
| Mix = $x_1$ | 75,8714 |
| Straordi = $x_2$ | 26,4286 |
| Mix*Straordi = $x_1x_2$ | -10,2286 |

Stima dei coefficienti con i regressori espressi mediante variabili codificate

| Term | Coef |
|---|---|
| Constant | 328 |
| Mix = $x_1$ | -11.07 |
| Straordi = $x_2$ | 13.21 |
| Mix*Straordi = $x_1x_2$ | -5.11 |

**Figura 6.39 Coefficienti di regressione**

dunque il modello assume la forma:

*Variabili non codificate*

$$y = 103.357 + 75.8714x_1 + 26.4286x_2 - 10.1186x_1x_2 \qquad \text{(e 6.5)}$$

*Variabili codificate*

$$y = 328 - 11.07x_1 + 13.21x_2 - 5.11x_1x_2 \qquad \text{(e 6.6)}$$

Nel caso in esame, il fattore mix produttivo è stato trattato come fattore qualitativo, mentre lo straordinario è un fattore di tipo quantitativo: questo implica che, nella $y = 103.357 + 75.8714x_1 + 26.4286x_2 - 10.1186x_1x_2$ (e 6.5), il regressore $x_2$ può assumere qualunque valore compreso tra 8 e 9 ore, mentre il regressore $x_1$ assume valore -1 se si vuole analizzare la risposta in presenza del mix (3,2,1) e valore +1 in presenza del mix (4,1,1). Nella

$y = 328 - 11.07x_1 + 13.21x_2 - 5.11x_1x_2$ (e 6.6), invece, ciascun regressore può essere espresso solo mediante variabili codificate.

L'equazione $y = 103.357 + 75.8714x_1 + 26.4286x_2 - 10.1186x_1x_2$ (e 6.5) rappresenta, sotto determinate ipotesi, il modello di simulazione creato e, quindi, l'azienda CO.PRO. Essa può essere utilizzata in sostituzione del modello di simulazione per avere una stima del numero di congelatori impostato ottenendo, dunque, un notevole risparmio di risorse in termini di lunghezza e numerosità di run della simulazione. Bisogna però sottolineare che l'equazione $y = 103.357 + 75.8714x_1 + 26.4286x_2 - 10.1186x_1x_2$ (e 6.5)può essere utilizzata:

- per tutti i modelli prodotti sulla prima linea, purchè nel turno lavorativo si producano due dei tre modelli processati su di essa;
- per la produzione di tutti i modelli che sulla seconda linea hanno lo stesso tempo di processamento del CH185 e del CH300;
- per la produzione di tutti i modelli che sulla terza linea hanno lo stesso tempo di processamento del CH350, purchè venga processato un solo congelatore grande standard per turno;
- solo per i mix produttivi (3,2,1) e (4,1,1);
- la risposta può essere valutata solo all'ottava ora, alla nona e in istanti intermedi ma mai in istanti di tempo $t < 8$ e $t > 9$.

Essendoci un solo fattore quantitativo, non è possibile ottenere il grafico della superficie di risposta, perché la sua costruzione richiede una stima della risposta in corrispondenza di un livello intermedio del fattore, che non è presente nel caso in cui questo sia di tipo qualitativo. Tuttavia è possibile plottare la $y = 103.357 + 75.8714x_1 + 26.4286x_2 - 10.1186x_1x_2$ (e 6.5) al variare dello straordinario tra 8 e 9 ore e fissando il livello del mix produttivo (Figura 6.40):

*Mix (3,2,1)* $\Rightarrow x_1 = -1$

$y = 27.4856 + 36.5472x_2$

*Mix (4,1,1)* $\Rightarrow x_1 = +1$

$y = 179.2284 + 16.31x_2$

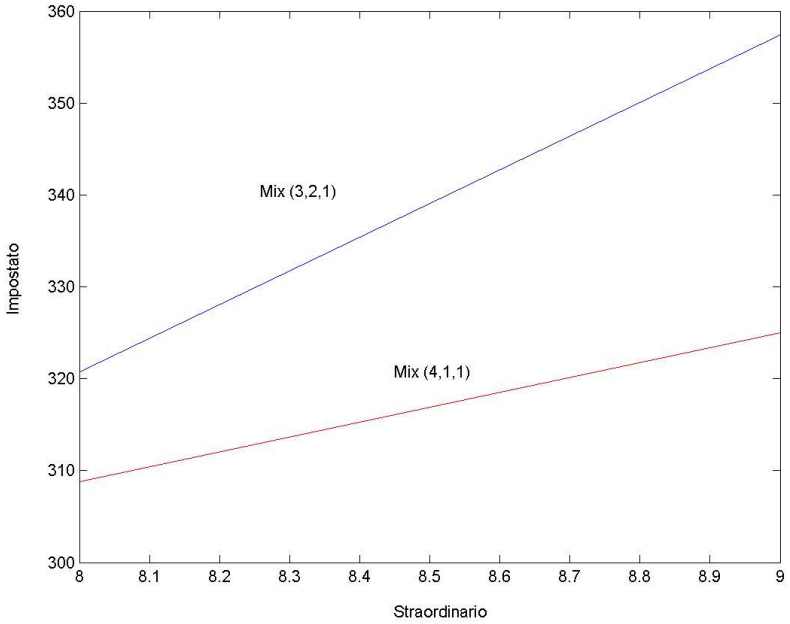



**Figura 6.40 Grafico del modello di regressione**

È facile notare che la Figura 6.40 è identica al grafico delle interazioni di Figura 6.35.

# 7. Bibliografia

[1] Aarts, E. H. L., Van Laarhoven, P. J. M., Lenstra, J. K. and Ulder, N. L. J. (1994) "A Computational Study of Local Search Algorithms for Job-Shop Scheduling," ORSA Journal on Computing, 6(2), Spring, 118-125.

[2] Aarts, E. H. L. and Lenstra, J. K. (eds) (1997) "Local Search in Combinatorial Optimization", Wiley, Chichester.

[3] Adams, J., Balas, E. and Zawack, D. (1988) "The Shifting Bottleneck Procedure for Job-Shop Scheduling", Management Science, March, 34(3), 391-401.

[4] Akers, S. B. (1956) "A Graphical Approach to Production Scheduling Problems", Operations Research, vol 4, 244-245.

[5] J.H. Ahmadi, R.H. Ahmadi, S. Dasu, C.S. Tang, "Batching and scheduling jobs on batch and discrete processors", Operations Research, 2000.

[6] A. Alan, B. Pritsker, "Modeling in erformance- Enhancing processes"; Operation Research (1997), Vol.45, No.6, pp 797-804;

[7] Alexopoulos G. S. Fishman; "Computational experience with the batch means method"; C. Proceedings of the 1997 Winter Simulation Conference ed. S. Andraddttir, K. J. Healy, D. H. Withers, and B. L. Nelson

[8] C. Alexopoulos, S.H Kim, "Output data analysis for simulation", Proceeding of the 2002 Winter Simulation Conference.

[9] C. Alexopoulos, A.F. Seila; "Output analysis for simulation", Proceedings of the 2000 Winter Simulation Conference J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

[10] Anant S. J. • Sheik M. (1998) "A State-Of-The-Art Review Of Job-Shop Scheduling Techniques"

[11] Anchordoguy, *"Breve storia del Keiretsu giapponese"*, in Harvard Business Review, Ed. Italiana – n. 5, 1995

[12] T.W. Anderson, D.A. Darling; " Asymptotic theory of certain goodness of fit criteria based on stochastic processes"; Annals of Mathematical Statistics (1952), 23: 193-212.

[13] M. Andersson e G. Olsson; "A simulation based decision support approach for operational capacity planning in a customer order driven assembly line" ;

Proceedings of the 1998 Winter Simulation Conference D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.;
[14] Applegate, D. and Cook, W. (1991) "A Computational Study of the Job-Shop Scheduling Problem", ORSA Journal on Computing, Spring, 3(2), 149-156.
[15] J. April, F. Glover, J. P. Kelly, M. Laguna, "Practical introduction to simulation optimization", Proceeding of the 2003 Winter Simulation Conference.
[16] M. Arakawa, M. Fuyuki, I. Inoue, "A simulation-based production scheduling method for minimizing the due-date-deviation";; International Transaction in Operational Research (2002), 9, pp.153-167;
[17] C.T. Baker, B. P. Dzielinski, "Simulation of simplified job shop", Management Science (1960) Vol.6, No.3, pp.311-323;
[18] Balas, E., and Vazacopoulos, A. (1998) "Guided Local Search with Shifting Bottleneck for Job-Shop Scheduling", Management Science, Feb, 44(2), 262-275.
[19] Balsamo S., R. Mirandola, "Modelli e metodi per la valutazione delle prestazioni di sistemi", Ed. SEU, 1996.
[20] J. Banks, J.S. Carson, B.L. Nelson, "Discrete-Event System Simulation", Ed.Prentice Hall, 1996.
[21] Baptiste, P., Le Pape, C. and Nuijten, W. P. M. (1995) "Constraint-Based Optimization and Approximation for Job-Shop Scheduling", AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems, 14th International Joint Conference on Artificial Intelligence, Montréal, Québec, Canada, Aug 19, pp. 5-16.
[22] Barnes, J. W. and Chambers, J. B. (1995) "Solving the Job Shop Scheduling Problem Using Tabu Search", IIE Transactions, vol 27, 257-263.
[23] M. Becker , A. L. Beylot, G. Damm ,W. Y. Thang; "Automatic run-time choice for simulation length in mimesis"; Rairo Recherche operationnelle/Operations Research, Vol. 33, No.1, 1999, pp. 93-115;
[24] Bettonvil, B. and J.P.C. Kleijnen, "Searching for important factors in simulation models with many factors: sequential bifurcation" (1997) European Journal of Operational Research, 96, no. 1, pp. 180-194
[25] P. J. Bickel, D. A. Freedman; "Some asymptotic theory for the bootstrap"; Annals of Statistics (1981) 9,pp. 1196-1217;
[26] Bierwirth, C., Mattfeld, D. C. and Kopfer, H. (1996) "On Permutation Representations for Scheduling Problems," Voigt, H. M. et al. (eds) PPSN'IV

Parallel Problem Solving from Nature, Springer-Verlag, Berlin, Heidelberg, Germany, pp. 310-318.

[27] Blackstone, J. H. Jr, Phillips, D. T. and Hogg, G. L. (1982) "A State of the Art Survey of Dispatching Rules for Manufacturing Job-Shop Operations", International Journal Of Production Research, Jan-Feb, vol 20, 27-45.

[28] Blazewicz, J., Domschke, W. and Pesch, E. (1996) "The Job-Shop Scheduling Problem: Conventional and New Solution Techniques", European Journal of Operational Research, 93(1), 23rd August, 1-33.

[29] Bonfiglioli R., "Lean Thinking alla maniera italiana", 2ª edizione, Franco Angeli, Milano, 2001

[30] Brucker, P. and Neyer, J. (1997) "Tabu Search for the Multi-Mode Job-Shop Problem", Operations Research Spektrum, vol 20, 21-28.

[31] J. M. Calvin, P. W. Glynn, M. K. Nakayama, "The semi-regenerative method of simulation output analysis", National Science Foundation, 2003.

[32] Carlier, J. and Pinson, E. (1994) "Adjustment of Heads and Tails for the Job-Shop Problem", European Journal of Operational Research, Oct 27, 78(2), 146-161.

[33] J.Carpenter, J. Bithell; "Bootstrap confidence intervals: when, which, what?A practical guide for medical statisticians"; Statistics in Medicine 2000; 19:1141-1164

[34] Caseau, Y. and Laburthe, F. (1995) "Disjunctive Scheduling with Task Intervals", LIENS Technical Report n° 95-25, Laboratoire d'Informatique de l'Ecole Normale Supérieure Département de Mathématiques et d'Informatique, 45 rue d'Ulm, 75230 Paris, France.

[35] V. Ceric, V. Hlupic, "Modelling a Solid-Waste Processing system by discrete event simulation"; The Journal of the Operational Research Society (1993), Vol.44, No.2, pp 107-114;

[36] Chandru V, Lee CY, Uzsoy R. – "Minimizing total completion time on batch processing machines" – International. Journal of Production Research 1993;31:2097-121.

[37] R. Chase, R. Jacobs, N. Aquilano "Operations management for competitive advantage", Mc Graw Hill, 2004.

[38] C.-H. Chen, H.-C. Chen, L. Dai; "A Gradient Approach For Smartly Allocating Computing Budget For Discrete Event Simulation"; Proceedings of the

1996 Winter Simulation Conference ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain;
[39] H.-C. Chen.,C.-H. Chen, L. Dai, E. Yücesan; "New Development Of Optimal Computing Budget Allocation For Discrete Event Simulation"; Proceedings of the 1997 Winter Simulation Conference ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson
[40] R. Chen, A. F. Seila, "Multivariate inference in stationary simulation using Batch Means", Proceeding of the 1987 Winter Simulation Conference
[41] R. C. H. Cheng; "Bootstrap methods in computer simulation experiments", Proceeding of the 1995 Winter Simulation Conference, C. Alexopoulos, W. R. Lilegdon, D. Goldsmann, eds.
[42] Cheng, C-C. and Smith, S. F. (1997) "Applying Constraint Satisfaction Techniques to Job Shop Scheduling", Annals of Operations Research, vol 70, 327-357.
[43] R. C.H. Cheng; "Analysis of simulation experiments by bootstrap resampling"; Proceeding of the 2001 Winter Simulation Conference B.A.Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer, eds.
[44] Chiarini A., "Total Quality Management", Franco Angeli, 2004
[45] S. Cho, "A distributed time-driven simulation method for enabling real-time manufacturing shop floor control", Computers & Industrial Engeneering (2005) , 49, 572-590;
[46] Chryssolouris, G., Wright, K. and Pierce, J. (1988) "A Simulator for Manufacturing Systems Based on Artificial Intelligence", Symposium on Advances in Manufacturing Systems Engineering, Nov 8 - Dec 2, Chicago,Ilinois, pp. 1-13.
[47] K. H. Concannon, K. I. Hunter e J. M. Tremble; "Simul8-Planner simulation-based planning and scheduling"; Proceedings of the 2003 Winter Simulation Conference S. Chick, P. J. Sánchez, D. Ferrin e D. J. morrice, eds;
[48] Dagli, C. H. and Sittisathanchai, S. (1995) "Genetic Neuro-Scheduler - A New Approach for Job-Shop Scheduling", International Journal of Production Economics, 41(1-3), 135-145.
[49] R.D. D'Agostino, H.A. Stephens, "Goodness-of-fit distribution", editors, M. Dekker, New York

[50] Dauzère-Pérès, S. and Paulli, J. (1997) “An Integrated Approach for Modeling and Solving the General Multiprocessor Job-Shop Scheduling Problem Using Tabu Search”, Annals of Operations Research, vol 70, 281-306.
[51] Davidor, Y., Yamada, T. and Nakano, R. (1993) “The Ecological Framework II: Improving GA Performance at Virtually Zero Cost”, ICGA’5 5th International Conference on Genetic Algorithms, pp. 171-176.
[52] Della Croce, F., Tadei, R. and Volta, G. (1995) “A Genetic Algorithm for the Job Shop Problem, Computers and Operations Research”, Jan, 22(1), 15-24.
[53] Dell’Amico, M. and Trubian, M. (1993) Applying Tabu “Search to the Job-Shop Scheduling Problem, Annals of Operations Research”, vol 41, 231-252.
[54] T.J.Di Ciccio, B. Efron, “Bootstrap Confidence Intervals”, Statistical science (1996) Vol.11, No.3, pp.189-228
[55] T.J.DiCiccio, J.P.Romano;”A review of Bootstrap confidence intervals”; Journal of the Royal Statistical Society. Series B (Methodological), Vol.50,No,3(1988), pp.338-354.
[56] ”, T. DiCiccio, R. Tibshirani;”Bootstrap Confidence intervals and Bootstrap approximation” Journal of the American Statistical association (1987) Vol.82 No. 397 Theory and Methods
[57] J. M. Donohue, “Experimental designs for simulation”, Proceeding of the 1994 Winter Simulation Conference.
[58] Drevna, M. and Kasales, C., “Introduction to Arena,” Series, 1993. Proceedings of the 1994 Winter Simulation Conference, M. Tew and S. Manivannan, Eds., IEE Publishers, Piscataway, NJ;
[59] L. DuPont, FJ Ghazvini – “A branch and bound algorithm for minimizing mean flow time on a single batch processing machine” – International Journal of Industrial Engineering, 1997.
[60] B. Efron, “Second thoughts on the Bootstrap” , Statistical science 2003, Vol.18, No.2, 135-140.
[61] B. Efron, R. Tibshirani; “Bootstrap methods for Standard Errors, confidence intervals, and other measures of statistical accuracy”; Statistical science(1986), Vol.1, No1, pp 54-75.
[62] P. Erto, “Probabilità e statistica per le scienze e l’ingegneria” seconda edizione, McGraw-Hill, 2004.

[63] Fisher, H. and Thompson, G. L. (1963) “Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules”, Muth, J. F. and Thompson, G. L. (eds) Industrial Scheduling, Prentice Hall, Englewood Cliffs, New Jersey, Ch 15, pp. 225-251.

[64] Fisher, M. L. and Rinnooy Kan, A. H. G. (1988) “The Design, Analysis and Implementation of Heuristics”, Management Science, March, 34(3), 263-265.

[65] G. A Fishman, “Principles of Discrete Event Simulation”, John Wiley, New York, 1978.

[66] Foo, S. Y., Takefuji, Y. and Szu, H. (1995) “Scaling Properties of Neural Networks for Job-Shop Scheduling”, Neurocomputing, 8(1), 79-91.

[67] Fox, M. S. and Sadeh, N. (1990) “Why Is Scheduling Difficult ? A CSP Perspective”, Aiello L. (ed) ECAI-90 Proceedings of the 9th European Conference on Artificial Intelligence, August 6-10, Stockholm, Sweden, pp. 754-767.

[68] M. Fuyuki, I. Inoue; “Due-date-conformance oriented production scheduling in make-to-order production on the basis of backward/forward hybrid simulation”; Journal of Japan Industrial Management Association (1995) 46, pp1. 144-151;

[69] Garey, M. R. and Johnson, D. S. (1979) “Computers and Intertractability: A Guide to the Theory of NPCompleteness”, W. H. Freeman, San Francisco.

[70] Giffler, B. and Thompson, G. L. (1960) “Algorithms for Solving Production Scheduling Problems”, Operations Research, 8(4), 487-503.

[71] A.K. Gupta, A.I. Sivakumar, “Multi-objective scheduling of two-job families on a single machine”, Omega 33 (2005) 399 – 405

[72] AK Gupta, AI. Sivakumar; “Simulation based multiobjective schedule optimization in semiconductor manufacturing.”; Proceedings of the 2002 Winter Simulation Conference (WSC 2002). San Diego, California: IEEE; 2002. p. 1862–70;

[73] Hanada, A. and Ohnishi, K. (1993) “Near Optimal Job-Shop Scheduling using Neural Network Parallel Computing”, IECON’93 Proceedings of the 19th Annual IEEE International Conference on Industrial Electronics, Control, and Instrumentation, Maui, Hawaii, Nov 15-19, vol 1, pp. 315-320.

[74] M. Hardy; “Actuarial application of the Bootstrap”; SOA 2004 New York Annual Meeting-session 72Ts, Bootstrap methods.

[75] Hefetz, N. and Adiri, I. (1982) “An Efficient Optimal Algorithm for the Two-Machines Unit-Time Job-Shop Schedule-Length Problem”, Mathematics of Operations Research, vol 7, 354-360.
[76] T. Hesterberg, S. Monaghan, D. S. Moore, A. Clipson e R. Epstein; “Bootstrap methods and permutation tests”; Companion chapter 18 to the practice of business statistics;
[77] O. Holthaus, O. Rosenberg, H. Ziegler; “Development and simulation of methods for scheduling and coordinating decentralized job shop using multi-computer systems”;
[78] S. Hood, P. D. Welch, “Experimental design issues in simulation with examples from semiconductor manufacturing”, Proceeding of the 1992 Winter Simulation Conference.
[79] Hopfield, J. J., and Tank, D. W. (1985) “Neural Computational of Decisions in Optimization Problems”, Biological Cybernetics, vol 52, 141-52.
[80] Y. Ikura, M. Gimple, “Scheduling algorithms for a single batch processing machine” – Operation Research Letters, 1986.
[81] Jain, A. S., and Meeran, S. (1998) “Job-Shop Scheduling Using Neural Networks”, International Journal of Production Research, 36(5), May, 1249-1272.
[82] Jackson, J. R. (1957) “Simulation Research on Job-Shop Production”, Naval Research Logistics Quarterly, vol 4, 287-295.
[83] Johnson N.J. “Modified t tests and confidence intervals for asymmetric populations”. Journal of the American Statistical Association(1978), 73, pp. 536-544
[84] Johnson, D. S., Aragon, C. R., McGeoch, L. A. and Schevon, C. (1989) “Optimization by Simulated Annealing: An Experimental Evaluation”; Part I, Graph Partitioning, Operations Research, 37(6), Nov-Dec, 865-892.
[85] K.Kelley; “The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals”; Educational and Psychological Measurement (1995), Vol.65 No.1.
[86] W. D. Kelton, “Designing simulation experiments”, Proceeding of the 1999 Winter Simulation Conference.

[87] W. David Kelton, R. R. Barton, “Experimental Design For Simulation”; Proceedings of the 2003 Winter Simulation Conference S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.
[88] W. D. Kelton, J. M. Charnes, “A comparison of confidence region estimators for multivariate simulation output”, Proceeding of the 1988 Winter Simulation Conference.
[89] W. David Kelton, R.P. Sadowski, D.A. Sadowski, “Simulation with Arena”, Mc Graw Hill, 2000.
[90] Kelton, W. D., D. A. Sadowski, and R. P. Sadowski, “Simulation with ARENA” , 1998, WCB/McGraw-Hill.
[91] J.P.C. Kleijnen; “Validation of models: statistical techniques and data availability”; Proceeding of the 1999 Winter Simulation Conference, P.A. Farrington, H.B. Nembhard, D.T. Sturrock e G.W. Evans eds.
[92] J P.C. Kleijnen, “Experimental designs for sensitivity analysis of simulation models”, Center of Economic Research, 2001.
[93] Kleijnen, J.P.C.; “Case study: statistical validation of simulation models.” European Journal of Operational Research(1995), 87, no. 1, pp. 21-34;
[94] J.P.C: Kleijnen, B. Bettonvil, W. Van Groenendaal; “Validation of trace driven simulation models: a novel regression test”; Management Science (1998), 44: 812-819.
[95] J. P.C. Kleijnen , R. C.H. Cheng e B. Bettonvill, ”Validation of trace.driven simulation models: Bootstrap tests”; Printed: May 8, 2001;
[96] J.P.C: Kleijnen; R.C.H. Cheng, B. Bettonvil; “Validation of trace-driven simulation models: more on Bootstrap tests”; Proceeding of the 2000 Winter Simulation Conference J.A. Joines, R.R. Barton e P. A. Fishwick, eds.
[97] Kleijnen, J.P.C., R.C.H. Cheng, and A.J. Feelders, “Bootstrapping and validation of metamodels in simulation” (1998), Proceedings of the 1998 Winter Simulation Conference, eds. D.J. Medeiros, E.F.
[98] Kleijnen, J.P.C. , G. Van Ham, and J. Rotmans, “ Techniques for sensitivity analysis of simulation models: a case study of the CO2 greenhouse effect”, (1992),Simulation, 58, no. 6, pp. 410-417
[99] Y.B.Kim, T.R. Willemain, J. Haddock, O. C. Runger; “The threshold bootstrap: A new approach to simulation output analysis”; Proceeding of the 1993

Winter Simulation Conference G.W.Evans, M. Mollaghesemi, E.C. Russell e W. E. Biles, eds.

[100] G.L.J. Kloppenburg, and F.L. Meeuwsen "Testing the mean of an asymmetric population: Johnson's modified t test revisited." Kleijnen, J.P.C. , Communications in Statistics, Simulation and Computation(1986),, 15, no. 3, pp.715-732.

[101] Kobayashi, S., Ono, I. and Yamamura, M. (1995) "An Efficient Genetic Algorithm for Job Shop Scheduling Problems", Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Francisco, CA, pp. 506-511.

[102] Krüger, K., Shakhlevich, N. V., Sotskov, Y. N. and Werner, F. (1995) "A Heuristic Decomposition Algorithm for Scheduling Problems on Mixed Graphs", Journal of the Operational Research Society, vol 46, 1481-1497.

[103] W. Kuehn, C. Draschba; "Simulation based job shop production analyser"; Proceedings 18th European Simulation Multiconference Graham Horton (c) SCS Europe, 2004;

[104] S. KUMAR e D. A. NOTTESTAD, "Capacity design: an application using discrete-event simulation and designed experiments", IIE Transactions (2006) 38, 729–736;

[105] Laguna, M., Barnes, J. W. and Glover, F. W. (1993) "Intelligent Scheduling with Tabu Search: An Application to Jobs with Linear Delay Penalties and Sequence-Dependent Setup Costs and Times", Journal of Applied Intelligence, vol 3, 159-172.

[106] Lamming, "Oltre la partnership, strategie per l'innovazione e la produzione snella", Ed. Italiana a cura di Capaldo, Passaro, Pastore, Raffa, Ed. CUEN, 1994

[107] A. M. Law, W. D. Kelton, "Simulation Modeling & Analysis", McGraw-Hill Int. Ed., 1991.

[108] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. and Shmoys, D. B. (1993) "Sequencing and Scheduling: Algorithms and Complexity", in Graves, S. C., Rinnooy Kan, A. H. G., Zipkin, P. H. (eds), Handbook in Operations Research and Management Science, Volume 4: Logistics of Production and Inventory, North Holland, Amsterdam.

[109] L. H. Lee, E. Peng Chew, S. Teng, D. Goldsman; "Optimal Computing Budget Allocation For Multi-Objective Simulation Models" , Proceedings of the

2004 Winter Simulation Conference R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

[110] E. LeGrande, "The development of a factory simulation system using actual operating data"; Management Tecnhnology(1963), Vol.3, No.1, pp 1-19;

[111] C.L. Li, C.Y. Lee – "Scheduling with agreeable release times and due dates on a batch processing machine" – European Journal of Operational Research, 1997.

[112] Lourenço, H. R. D. and Zwijnenburg, M. (1996) "Combining the Large-Step Optimization with Tabu-Search: Application to the Job-Shop Scheduling Problem," in Osman, I. H. and Kelly, J. P. (eds) Meta-heuristics: Theory and Applications, Kluwer Academic Publishers, Boston, MA, USA, Chapter 14, pp. 219-236.

[113] P. S. Mahajan, R. G. Ingalls; "Evaluation Of Methods Used To Detect Warm-Upperiod In Steady State Simulation"; Proceedings of the 2004 Winter Simulation Conference R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

[114] J. H. Marvel, M. A. Schaub e G. Weckman; "Validating the capacity planning process and flowline product sequencing through simulation analysis"; Proceedings of the 2005 Winter Simulation Conference M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

[115] Matsuo, H., Suh, C. J. and Sullivan, R. S. (1988) "A Controlled Search Simulated Annealing Method for the General Job-Shop Scheduling Problem", Working Paper #03-04-88, Graduate School of Business, The University of Texas at Austin, Austin, Texas, USA.

[116] Mattfeld, D. C., Bierwirth, C. and Kopfer, H. (1998) "A Search Space Analysis of the Job Shop Scheduling Problem", to appear in Annals of Operations Research.

[117] McMahon, G. B. and Florian, M. (1975) "On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness", Operations Research, May-June, 23(3), 475-482.

[118] G. Metan e I. Sabuncuoglu; "A simulation based learning mechanism for scheduling systems with continuous control and update structure" ; Proceedings of the 2005 Winter Simulation Conference M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

[119] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) “Equation of State Calculations by Fast Computing Machines”, The Journal of Chemical Physics, 21(6), June, 1087-1092.
[120] Douglas C. Montgomery, “Design and analysis of experiments”, fifth edition, John Wiley & Sons.
[121] Morgan, B. J. T. “Elements of Simulation”, Chapman & Hall, London 1994.
[122] R. Mosca, P. Giribone, G. Guglielmone, “Optimal Lenght in O.R. Simulation Experiments of Large Scale Production System”, International Symposium “Applied Modelling and Simulation”, Parigi, Luglio 1982
[123] R. Mosca, P. Giribone, “Teoria degli esperimenti e simulazione”, Università di Genova, 1985.
[124] R. Mosca, L. Cassettari, R. Revetria e G. Magro; “Simulation as support for production planning in small and medium enterprise: a case study”; Proceedings of the 2005 Winter Simulation Conference M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.;
[125] Nakano, R. and Yamada, T. (1991) “Conventional Genetic Algorithm for Job-Shop Problems”, in Kenneth, M. K. and Booker, L. B. (eds) Proceedings of the 4th International Conference on Genetic Algorithms and their Applications, San Diego, USA, pp. 474-479.
[126] M.K.. Nakayama; “Two-stages stopping procedures based on standardized time series”; Computer and information science department , new jersey institute of technology Newark,NJ 07102
[127] M. K. Nakayama, “Simulation output analysis”, Proceeding of the 2002 Winter Simulation Conference.
[128] M. K. Nakayama – “Simulation output analysis” – Proceeding of the 2002 Winter Simulation Conference.
[129] Nowicki, E. and Smutnicki, C. (1996) “A Fast Taboo Search Algorithm for the Job-Shop Problem”, Management Science, 42(6), 797-813.
[130] Nuijten, W. P. M. and Le Pape, C. (1998) “Constraint-Based Job Shop Scheduling with ILOG SCHEDULER”, Journal of Heuristics, March, 3(4), 271-286.
[131] Palaniswami S., Jenicke L., “A Knowledge-based Simulation System for Manufacturing Scheduling” International Journal of Operations & Production Management, (1992) Vol. 12 No. 11. pp. 4-14;

[132] Pegden, C., R. Sadowski, and R. Shannon., “Introduction to Simulation Using SIMAN, 2nd ed.”, 1995. McGraw-Hill, Singapore.
[133] Perregaard, M. and Clausen, J. (1995) “Parallel Branch-and-Bound Methods for the Job-Shop Scheduling Problem”, Working Paper, University of Copenhagen, Copenhagen, Denmark.
[134] Pesch, E., Tetzlaff, U. A. W. (1996) “Constraint Propagation Based Scheduling of Job Shops”, INFORMS Journal on Computing, Spring, 8(2), 144-157.
[135] Peterson, C. and Anderson, J. R. (1987) “A Mean Field Theory Learning Algorithm for Neural Networks”, Complex Systems, vol 1, 995-1019.
[136] Pidd M., “Computer Simulation in Management Science”, J.Wiley, 1998.
[137] Pinson, E. (1988) "Le Problème de Job-Shop, Thèse d’État", L’Université Pierre et Marie Curie, Paris VI, France. (In French).
[138] Ramudhin, A. and Marier, P. (1996) “The Generalized Shifting Bottleneck Procedure”, European Journal of Operational Research, vol 93, 34-48.
[139] Reeves, C. R. (1993) “Evaluation of Heuristic Performance”, in Reeves, C. R. (ed) Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publications, Osney Mead, Oxford, England, chapter 7, pp. 304-315. (Re-issued 1995 by McGraw-Hill, London.)
[140] Ripley, B.D. “Stochastic Simulation”, Ed. John Wiley, New York 1987.
[141] R. Roy, “Scheduling and control, performance measures and discrete event simulation”; The Journal of the Operational Research Society (1998), Vol.49, No.2, pp 151-156;
[142] A. Ruiz-Torres, J. e K. Nakatani; “Application of real-time simulation to assign due dates on logistic-manufacturing networks”; Proceedings of the 1998 Winter Simulation Conference D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.
[143] Sabuncuoglu, I. and Bayiz, M. (1997) “A Beam Search Based Algorithm for the Job Shop Scheduling Problem”, Research Report: IEOR-9705, Department of Industrial Engineering, Faculty of Engineering, Bilkent University, 06533 Ankara, Turkey (to appear in the European Journal of Operational Research).
[144] Sadeh, N., Nakakuki, Y. and Thangiah, S. R. (1997) “Learning to Recognise (Un)Promising Simulated Annealing Runs: Efficient Search Procedures for Job-

Shop Scheduling and Vehicle Routing", Annals of Operations Research 75, 189-208.
[145] D. Sadowski, V. Bapat, "The ARENA production family: enterprise modeling solutions", Proceeding of the 1999 Winter Simulation Conference.
[146] S. M. Sanchez, "ABC's of output analysis", Proceeding of the 1999 Winter Simulation Conference.
[147] R.G. Sargent; "Verification and validation of simulation models" Proceedings of the 2005 Winter Simulation Conference M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.
[148] S. Sawyer ; "Resampling Data:Using Bootstraps"; Washington University , March 12, 2005
[149] Schmidt, J. W., and R. E. Taylor, "Simulation and Analysis of Industrial Systems", Richard D. Irwin, Homewood, III 1970.
[150] Schriber, T.J., "The Nature and Role of Simulation in the Design of Manufacturing Systems", (1987) Simulation in CIM and Artificial Intelligence Techniques, Ed. Retti, J., and K. E. Wichmann, Society of Computer Simulation, 5-18;
[151] V. Selladurai P. Aravindan S.G. Ponnambalam e A. Gunasekaran, "Dynamic simulation of job shop scheduling for optimal performance"; International Journal of Operations & Production Management, (1995) Vol. 15 No. 17. pp. 106-120;
[152] V. Sergi, "Produzione assistita da calcolatore", Ed. Cues, 1998.
[153] R.E. Shannon, "Introduction to the art and science of simulation", Proceeding of the 1998 Winter Simulation Conference.
[154] C-C. Shen, "Discrete-Event Simulation On The Web" 1997 IEE;
[155] AI Sivakumar; "Multiobjective dynamic scheduling using discrete event simulation."; International Journal of Computer Integrated Manufacturing (2001);14(2):154–67;
[156] Jeffrey S. Smith, David T. Sturrock Sanjay E. Ramaswamy, "Discrete Event Simulation For Shop Floor Control", Proceedings of the 1994 Winter Simulation Conference ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila;
[157] L. Soliani "Fondamenti di statistica applicata", Edizione settembre 2002.
[158] N. M. Steiger, "ASAP3: a batch means procedure for Stedy-State simulation analysis", University of Maine.

[159] N. M. Steiger, J. M. Wilson, “Experimental performance evaluation of batch means procedures for simulation output analysis”, Proceeding of the 1994 Winter Simulation Conference.
[160] N. M. Steiger, J. M. Wilson, “An improvement batch means for simulation output analisysis”, University of Maine.
[161] Storer, R. H., Wu, S. D. and Park, I. (1993) “Genetic Algorithm in Problem Space for Sequencing Problems”, in Fandel, G., Gulledge, T. and Jones, A. (eds), Operations Research in Production Planning and Control: Proceedings of a Joint US/German Conference, Springer Verlag, Berlin, Heidelberg, pp. 584-597.
[162] C.S. Sung, S.H. Yoon – “Minimizing maximum completion time in a two batch-processing machine flowshop with dynamic arrivals allowed” – Engineering Optimization, 1997.
[163] Swart, W., and L. Donno “Simulation Modeling Improves Operations, Planning, and Productivity”, 1981.
[164] Tamaki, H. and Nishikawa, Y. (1992) “A Parallelled Genetic Algorithm Based on a Neighbourhood Model and its Application to the JobShop Scheduling”, in Männer, R. and Manderick, B. (eds) PPSN’2 Proceedings of the 2nd International Workshop on Parallel Problem Solving from Nature, Brussels, Belgium, pp. 573-582.
[165] F. Turco, “Principi generali di progettazione degli impianti industriali”, Ed. CLUP, 1990.
[166] N. Ueno, S. Sotojima, J. Takeda; “Simulation-Based Approach to Design a Multi-Stage Flow-Shop in Steel Works”, IEEE 1991;
[167] Urgeletti, Tinarelli, “La gestione delle scorte nelle imprese commerciali e di produzione – EOQ, MRP, JIT”, ETASLIBRI, 1994
[168] R. Uzsoy – “Scheduling batch processing machines with incompatible job families” – International Journal of Production Research, 1995.
[169] B. S. Vaidyanathan, D. M. Miller e Y. H. Park, “Application Of Discrete Event Simulation In Production Scheduling”, Proceedings of the 1998 Winter Simulation Conference D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.
[170] R.L. Van Horn; “Validation of simulation results”; Management Science, Vol.17 No.5, Theory Series (1971), pp247-258.

[171] Vaessens, R. J. M., Aarts, E. H. L. and Lenstra, J. K. (1996) "Job Shop Scheduling by Local Search", INFORMS Journal on Computing, vol 8, 302-317.
[172] S.Vincent, "Input data analysis" Handbook of simulation, edited by J. Banks, Wiley, New York;
[173] Wang, W. and Brunn, P. (1995), "Production Scheduling and Neural Networks", in Derigs, U., Bachem, A. and Drexl, A. (eds.), Operations Research Proceedings 1994, Springer-Verlag, Berlin, pp. 173-178.
[174] Watanabe, T., Tokumaru, H. and Hashimoto, Y. (1993) "Job-Shop Scheduling using Neural Networks", Control Engineering Practice, Dec, 1(6), 957-961.
[175] E. F. Watson, D. J. Medeiros e R. P. Sadowski, "A Simulation-Based Backward Planning Approach For Order-Release"; Proceedings of the 1997 Winter Simulation Conference ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson;
[176] G. Weiller, "Il controllo del processo di produzione", Franco Angeli, 1996.
[177] Werner, F. and Winkler, A. (1995) "Insertion Techniques for the Heuristic Solution of the Job-Shop Problem", Discrete Applied Mathematics, 58(2), 191-211.
[178] Williamson, D. P., Hall, L. A., Hoogeveen, J. A., Hurkens, C. A. J., Lenstra, J. K., Sevast'janov, S. V. and Shmoys, D. B. (1997) "Short Shop Schedules", Operations Research, March - April, 45(2), 288-294.
[179] Womack J.P., Jones D.T., Roos D., "La macchina che ha cambiato il mondo", Ed. BUR, 6ª RISTAMPA, Ottobre 2000
[180] Yamada, T. and Nakano, R. (1996c) "A Fusion of Crossover and Local Search", ICIT'96 IEEE International Conference on Industrial Technology, Shanghai, China, Dec 2-6, pp. 426-430.
[181] J. Yang ; T.-S Chang ; "Multiobjective scheduling for IC sort and test with a simulation testbed"; IEEE transactions on semiconductor Manufacturing (1998), vol. 11,no2,pp.181-231 (19), pp. 304-315;
[182] E. Yücesan; " Nonparametric techniques in simulation analysis: a tutorial"; Proceeding of the 1994 Winter Simulation Conference J.D. Tew, S. Manivannan, D.A. Sadowski e A. F. Seila, eds.

[183] E. Yücesan, C.-H. Chen, J. L. Snowdown e M. Charnes, eds; “The role of simulation in advanced planning and scheduling”; K. Musselman; Proceedings of the 2002 Winter Simulation Conference [
[183] M. K. Nakayama; Selecting The Best System In Steady-State Simulations Using Batch Means; Proceedings of the 1995 Winter Simulation Conference ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman
[184] Zhang, H.-C. and Huang, S. H. (1995) Applications of Neural Networks in Manufacturing: A State-of-the-Art Survey, International Journal of Production Research, 33(3), 705-728.