

# **RAPPORTO SUL MODELLO SIMULATIVO DEL PROCESSO PRODUTTIVO**

## **1. Contesto aziendale e processo produttivo**

L'industria alimentare rappresenta uno dei comparti più rilevanti del settore secondario in Italia, caratterizzandosi per l'integrazione tra tradizione artigianale, innovazione tecnologica e capacità di penetrazione sui mercati internazionali. All'interno di questo panorama si colloca Vicenzi SPA virgola in realtà emblematica del settore dolciario italiano.

Fondata nel 1905 da Matilde Vicenzi a San Giovanni Lupatoto, in provincia di Verona l'azienda ha saputo trasformarsi nel corso del tempo dal laboratorio artigianale a gruppo industriale di dimensioni internazionali. La crescita è stata accompagnata da scelte strategiche rilevanti come l'acquisizione dei marchi "Grisbi" e "Mr.Day" virgola che hanno ampliato l'offerta e consolidato la presenza sui mercati nazionali e globali.

Oggi il gruppo conta tre stabilimenti produttivi, impiega oltre 360 addetti e realizza un fatturato superiore a 160 milioni di euro, esportando in quasi cento paesi. La sua missione si fonda su tre pilastri: la valorizzazione della tradizione dolciaria italiana, l'investimento continuo in ricerca e sviluppo, e l'impegno verso la sostenibilità ambientale e sociale.

Dal punto di vista produttivo, Vicenzi si distingue per la capacità di unire processi altamente automatizzati con il mantenimento di ricette e standard qualitativi tipici dell'artigianalità. Le fasi principali del ciclo produttivo comprendono:

- La selezione delle materie prime, con particolare attenzione alla tracciabilità e all'origine certificata;
- La lavorazione e preparazione degli impasti, attraverso linee automatizzate che garantiscono uniformità e qualità;
- La cottura, raffreddamento e confezionamento, effettuati con macchinari avanzati e packaging sostenibili;
- Il controllo qualità e la logistica, fondamentali per garantire sicurezza alimentare, efficienza distributiva e rispetto delle normative internazionali.

Questa combinazione di tradizione, innovazione e organizzazione rende l'azienda un caso ideale per sviluppare un modello simulativo del processo produttivo, utile sia in chiave didattica che gestionale.

## **2. Codice Python sviluppato**

Alla fine della rappresentazione in forma semplificata il processo produttivo è stato sviluppato un modello in linguaggio Python che consente di simulare la produzione di un lotto composto da più tipologie di prodotti.

Il codice si fonda su alcuni elementi chiave:

- La generazione causale delle quantità da produrre per ciascun prodotto;
- L'assegnazione di parametri operativi variabili (tempo unitario di produzione, capacità giornaliera per-prodotto e capacità complessiva dell'impianto);

- Il calcolo del makespan, inteso come tempo minimo necessario, espresso in giornate intere per completare la produzione del lotto;
- La produzione di un report leggibile, utile a interpretare lo scenario e i risultati ottenuti.

Il codice è stato scritto, facendo uso esclusivamente di librerie standard già incluse nell'ambiente di programmazione. Questa scelta risponde all'esigenza di mantenere il modello leggero, portabile e facilmente eseguibile anche su sistemi privi di pacchetti esterni.

In particolare, sono state utilizzate le seguenti librerie:

- **Random**

La libreria è stata impiegata per la generazione di numeri casuali, sia interi sia decimali. Essa è fondamentale per introdurre variabilità nella quantità da produrre e nei parametri operativi (tempi unitari, capacità giornaliere). L'uso della funzione Random (seed) consente inoltre di fissare un seme iniziale e ottenere così risultati ripetibili, caratteristica utile per verificare e replicare la simulazione;

- **Math**

All'interno del modello, la libreria math è stata utilizzata per l'impegno della funzione ceil, che permette di arrotondare all'intero superiore i valori decimali. Tale operazione è necessaria nel calcolo del makespan, in quanto i risultati devono essere espressi in giornate operative intere, non in frazioni di giorno.

- **Typing**

Questa libreria non incide direttamente sul calcolo, ma è stata utilizzata per migliorare la leggibilità del codice. In particolare, i suoi strumenti consentono di specificare il tipo di variabile e valori restituiti dalle funzioni, ad esempio Dict, List, Tuple. Tali annotazioni hanno finalità documentative e didattiche.

L'uso di sole librerie standard dimostra che anche con strumenti di base è possibile realizzare una simulazione efficace e chiara, non avendo quindi la necessità di ricorrere a framework complessi o pacchetti esterni.

## **Struttura del codice**

Il programma è stato organizzato seguendo un approccio modulare, al fine di rendere il codice di facile lettura, modificabile ed estendibile. Ogni parte della simulazione è stata isolata in una funzione dedicata, con l'obiettivo di separare i compiti e ridurre la complessità complessiva.

La struttura logica del codice può essere suddivisa in tre blocchi principali:

- **Definizione dei parametri di base**

Nella parte iniziale del file sono stati definiti i valori costanti e gli intervalli utilizzati per la generazione dei dati (ad esempio quantità minime e quantità massime, i range dei tempi unitari e delle capacità giornaliere). Questa sezione ha la funzione di rendere il codice configurabile: modificando tali valori è possibile simulare scenari diversi senza alterare la logica del programma.

- **Definizione delle funzioni**

Le funzioni genera\_quantita, genera\_parametri, calcola\_makespan e stampa\_report racchiudono la logica del modello, ciascuna con un compito specifico. La scelta di utilizzare

funzioni distinte garantisce chiarezza e facilita eventuali riutilizzi e estensioni future (ad esempio l'aggiunta di nuovi vincoli o indicatori di performance).

### **Blocco principale di esecuzione**

L'ultima parte del file è racchiusa all'interno della condizione `if __name__ == "__main__":`:

Questo costrutto è tipico di Python e serve a distinguere due casi:

- se il file viene eseguito direttamente, il blocco viene avviato e la simulazione parte;
- se il file viene importato con modulo in un altro programma, il blocco non viene eseguito automaticamente.

All'interno di questo blocco si susseguono quattro passaggi logici:

1. Generazioni casuale delle quantità da produrre;
2. Generazione casuale dei parametri operativi;
3. Calcolo del makespan sulla base dei dati ottenuti;
4. Stampa del report finale, contenente lo scenario e i risultati della simulazione.

Questa organizzazione lineare, con parametri iniziali, funzioni modulari e blocco principale, rende il codice semplice da leggere e da spiegare.

### **Funzioni Implementate**

Il programma è stato suddiviso in più funzioni, ognuna con un compito ben definito. Questo approccio consente di isolare le singole responsabilità e facilita sia la lettura che la manutenzione del codice. Le funzioni implementate sono:

#### **`genera_quantita`**

Questa funzione si occupa di definire la domanda produttiva che prende in input: l'elenco dei prodotti, un intervallo di quantità ammissibili e il generatore casuale, dando in output: un dizionario che associa a ciascun prodotto la quantità da produrre. La logica su cui si basa è: per ogni prodotto viene estratto un valore intero compreso nell'intervallo stabilito.

In questo modo è possibile simulare scenari diversi ad ogni esecuzione, mantenendo tuttavia la ripetibilità dei risultati grazie al seme fissato nel generatore casuale.

#### **`genera_parametri`**

Questa funzione definisce i vincoli operativi del sistema produttivo. Prende in input l'elenco dei prodotti, gli intervalli dei tempi unitari, delle capacità per prodotto e della capacità complessiva giornaliera, fornendo in output: come primo risultato un dizionario che per ogni prodotto contiene il tempo medio di lavorazione per unità (`t_unit`) e la capacità massima giornaliera (`cap_giorno`); come secondo risultato il valore della capacità complessiva giornaliera espressa in ore (`cap_tot_ore_giorno`). I valori estratti casualmente entro i range forniti, così fa simulare condizioni operative differenti. Questa funzione permette quindi di introdurre nel modello la variabilità tipica dei sistemi reali, legata a fattori tecnologici, organizzativi e strutturali.

#### **`calcola_makespan`**

È la funzione che esegue il calcolo vero e proprio del tempo minimo necessario per completare il lotto. In input abbiamo le quantità da produrre, i parametri operativi e la capacità complessiva giornaliera. In output otteniamo un dizionario che riporta: il tempo totale di lavorazione (espresso in ore); i giorni minimi richiesti dal vincolo di orari giornalieri dell'impianto; i giorni minimi richiesti dal vincolo di capacità per prodotto; il makespan finale, espresso in giornate intere. Il makespan viene determinato prendendo il massimo tra i due vincoli (globale e per prodotto) arrotondando all'intero superiore. In questo modo si ottiene una stima sintetica ma significativa della durata complessiva del processo produttivo.

### **stampa\_report**

Questa funzione allo scopo di rendere i risultati leggibili e facilmente interpretabili. In input abbiamo i nomi dei prodotti, le quantità generate, i parametri operativi, la capacità complessiva e i risultati calcolati. In output riceviamo una stampa ordinata a schermo. Analizzando nello specifico, vengono prima e riportati i parametri dello scenario, poi le quantità e i tempi richiesti per ogni prodotto, e infine i risultati globali della simulazione (ore totali, giorni minimi, makespan). Grazie a questa funzione, il modello non restituisce solo numeri grezzi, non vero e proprio report di sintesi.

## **3.Processo di sviluppo del codice**

Il percorso di sviluppo del modello si è articolato in più fasi:

### **1. Analisi del contesto produttivo:**

È stato descritto il processo produttivo tipico del settore dolciario, individuando le fasi principali (approvvigionamento, lavorazione, confezionamento, logistica) e i relativi vincoli.

### **2. Definizione degli obiettivi del modello:**

L'obiettivo principale è stato stimare il tempo necessario per completare un lotto di produzione con più prodotti, rispettando vincoli di capacità globali che specifici per-prodotto.

### **3. Progettazione della struttura del codice:**

Il programma è stato suddiviso in funzioni modulari, ciascuna con un compito chiaro: generazione delle quantità, Generazione dei parametri, calcolo del makespan, stampa del report.

### **4. Implementazione e commento:**

Ogni funzione è stata sviluppata con un linguaggio semplice e arricchita di commenti, così da risultare leggibile anche ad un pubblico non esperto di programmazione.

### **5. Esecuzione e analisi dei risultati:**

Il programma è stato testato fissando un seme casuale, così da garantire la ripetibilità dei dati. I risultati hanno mostrato come, anche in un modello semplificato, il makespan sia fortemente influenzato dal vincolo del tempo complessivo disponibile giornalmente.

## **4.Considerazioni finali**

Il modello sviluppato rappresenta un primo passo verso la simulazione dei processi produttivi nel settore alimentare. La sua semplicità costituisce al tempo stesso un limite e un punto di forza: se da un lato non include variabili quali scarti, tempi di setup o manutenzioni, dall'altro permette di comprendere con immediatezza il legame tra quantità da produrre, capacità operativa e durata complessiva.

Tra i possibili risvolti futuri si segnalano:

- L'introduzione di ulteriori vincoli e parametri, per rendere la simulazione più vicina alla realtà;
- l'elaborazione di una schedulazione giorno per giorno, per visualizzare la distribuzione dei lotti;
- l'integrazione di tecniche di ottimizzazione e algoritmi di scheduling, volti a individuare soluzioni più efficienti.

In conclusione, il lavoro mostra come l'unione di competenze economiche e informatiche posso produrre strumenti utili per analisi e decisioni in ambito industriale, confermando il potenziale della simulazione come supporto formativo e gestionale.