

# ***Robust Supply Chain Networks with Monte Carlo Simulation: Implementazione ed Applicazione***

## ***1. Obiettivo***

Costruire una semplice metodologia di Supply Chain Network Design che tenga conto della fluttuazione della domanda.

## ***2. Scenario***

L'ottimizzazione della supply chain sfrutta al meglio l'analisi dei dati per trovare una combinazione ottimale di fabbriche e centri di distribuzione per soddisfare la domanda dei clienti.

In molti software e soluzioni sul mercato, la struttura alla base è formata da un modello di programmazione lineare. Alcuni di questi modelli trovano la giusta allocazione delle fabbriche per soddisfare la domanda e ridurre al minimo i costi assumendo una domanda costante.

Quello che, a questo punto, ci si chiede è cosa succede quando la domanda è *fluttuante*.

La rete potrebbe perdere robustezza, soprattutto quando si ha una stagionalità molto alta della domanda (e-commerce, cosmesi, fast fashion), quindi, sostanzialmente, quando si hanno dei periodi ad alta domanda e periodi a bassa domanda.

Quello che ci si propone è di costruire una semplice metodologia per definire una Supply Chain Network robusta usando la simulazione Monte Carlo con python.

## ***3. Supply Chain Network Design***

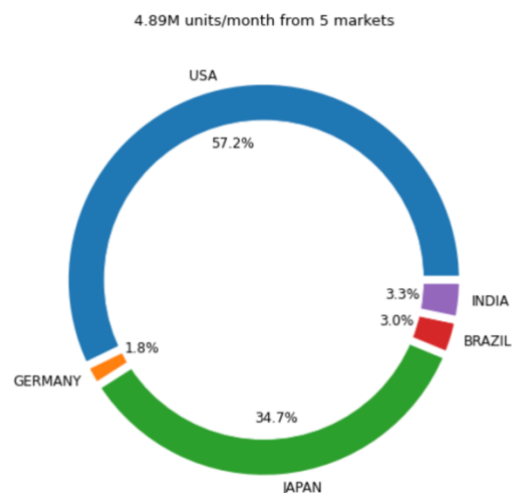
Il capo della Supply Chain Management di un'azienda internazionale manifatturiera vuole ridefinire la Supply Chain Network per i prossimi 5 anni.

Quindi, si suppongono le seguenti caratteristiche per il mercato.

NOTA: Tutti i grafici esposti vengono generati in Python sfruttando i dati a disposizione.

### **1) Domanda.**

La domanda dei clienti proviene da 5 diversi mercati (Brasile, USA, Germania, India e Giappone).



*Figura 1: Market Demand*

## 2) Supply Capacity.

Si possono aprire fabbriche nei cinque mercati. È possibile scegliere tra strutture a bassa e alta capacità.

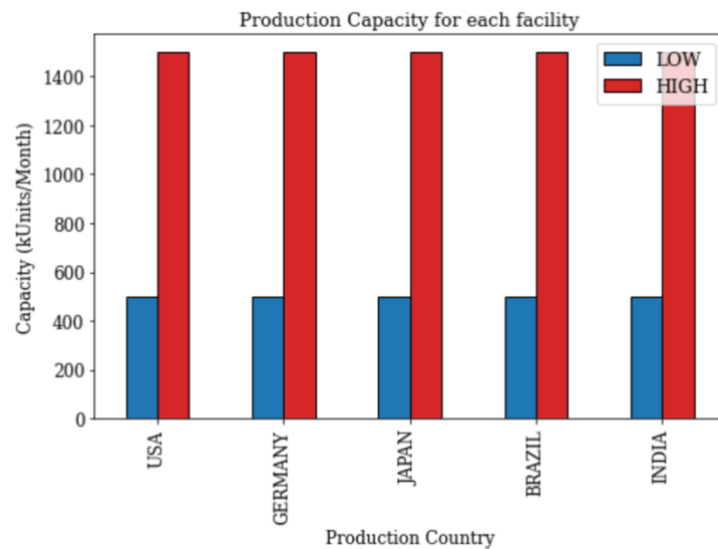


Figura 2: Capacità Fabbriche

Per esempio, si può aprire una fabbrica negli USA a bassa capacità che va a produrre 500.000 unità per mese.

## 3) Fixed Costs

Costi fissi per i vari paesi da affrontare se si apre una fabbrica (Elettricità, etc.).

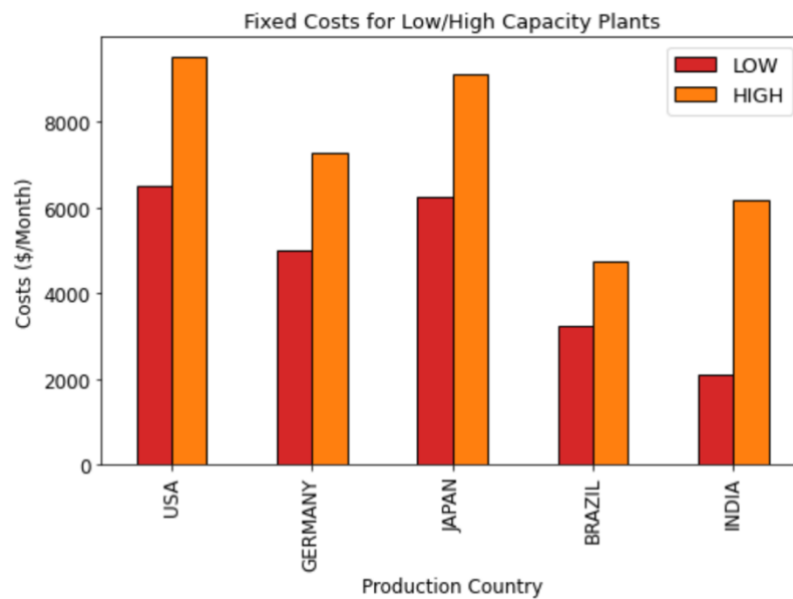


Figura 3: Costi Fissi Fabbriche

#### 4) Variable Costs

Costo totale di produzione = Costi Variabili + Costi Fissi.

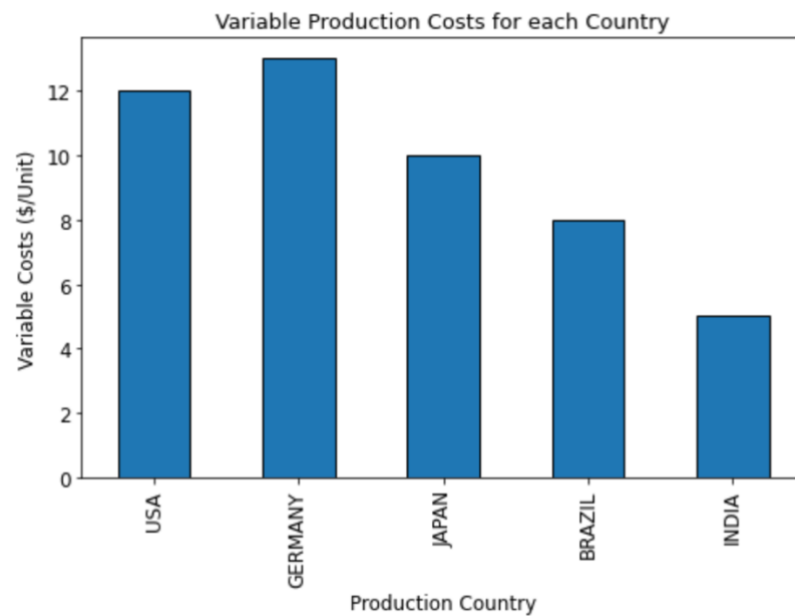


Figura 4: Costi Variabili

#### 5) Freight Costs

Costo per spedire un container dal paese X al paese Y.

	USA	GERMANY	JAPAN	BRAZIL	INDIA
USA	0	12250	1100	16100	8778
GERMANY	13335	0	8617	20244	10073
JAPAN	15400	22750	0	43610	14350
BRAZIL	16450	22050	28000	0	29750
INDIA	13650	15400	24500	29400	0

Figura 5: Costi per il trasporto (\$/container)

## 6) Total Costs

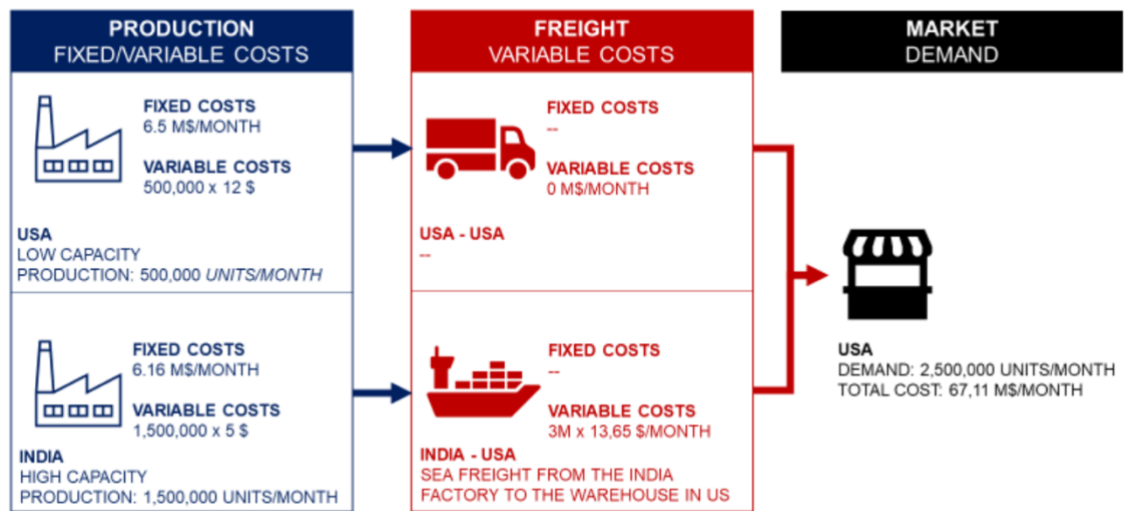


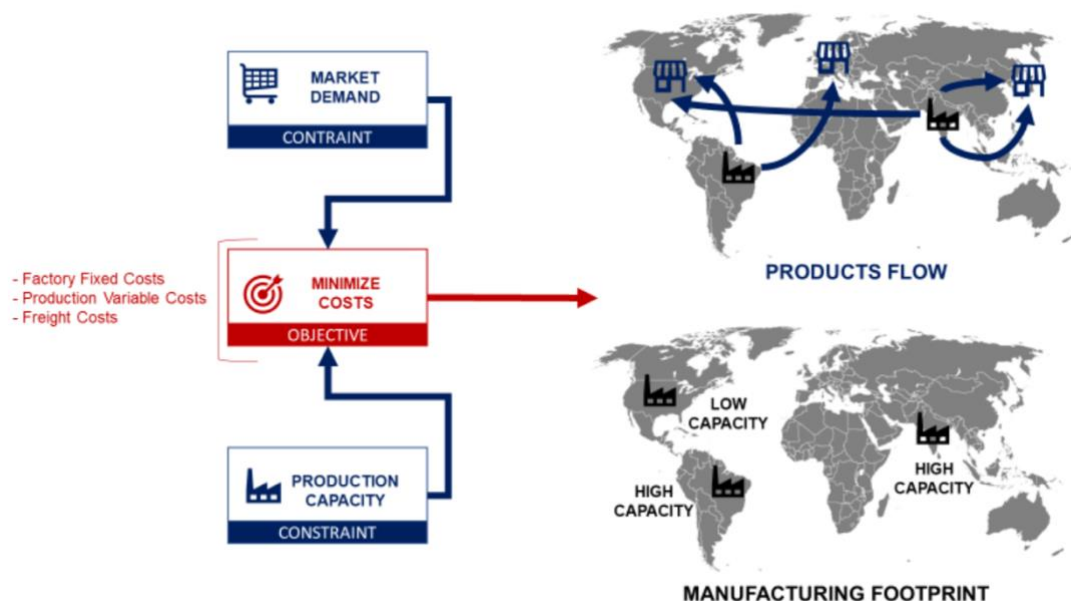
Figura 6: Costi totali per produrre e spedire

A questo punto, ci si pone la seguente domanda:

Basandoci sulla domanda del mercato, dove si andrà ad aprire le fabbriche?

## 4. Modello di programmazione Lineare

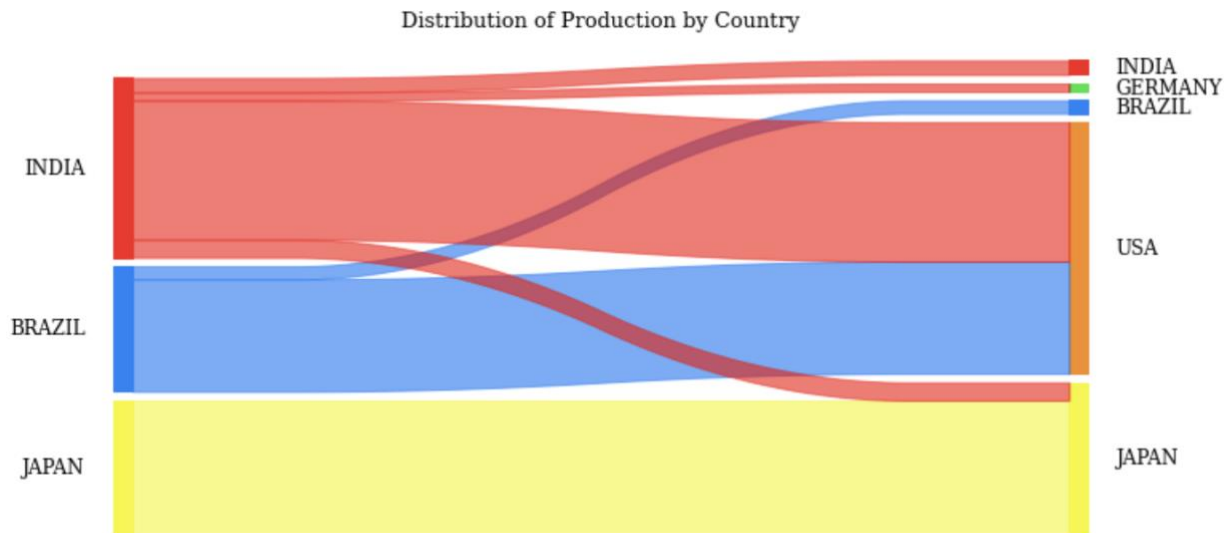
Quello che si va a definire è un classico problema di programmazione lineare con una funzione obiettivo, vincoli e parametri.



Si suppongono le seguenti aperture come Soluzione Iniziale:

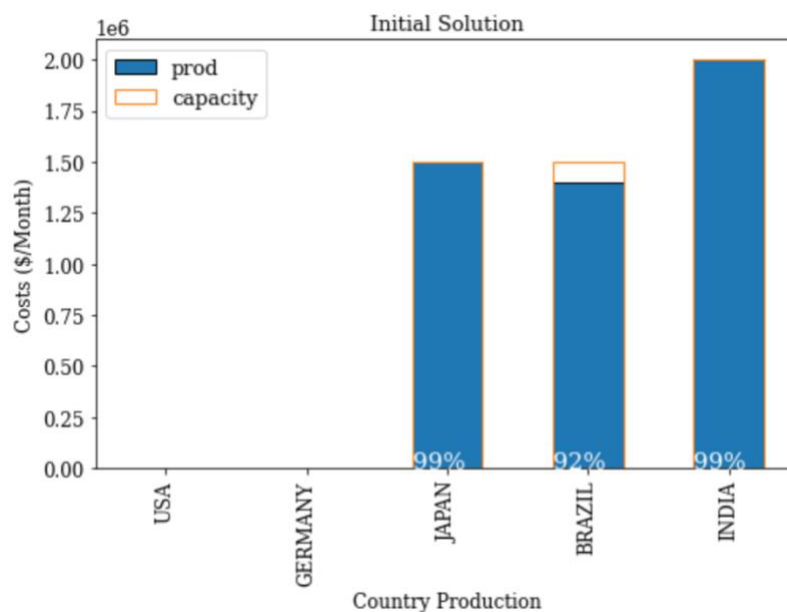
Quattro sedi in Brasile, Stati Uniti, Giappone e India. Fatta eccezione per il Brasile, tutte queste località sono impianti ad alta capacità.

Quante unità sono prodotte dalla fabbrica YYY per il Market XXX?



La fabbrica giapponese produce solo per il mercato locale, mentre Brasile e India sono principalmente guidate dalla domanda di esportazione.

Ora, quale potrebbe essere l'impatto a fronte di un +10 in Giappone e +20 in USA?

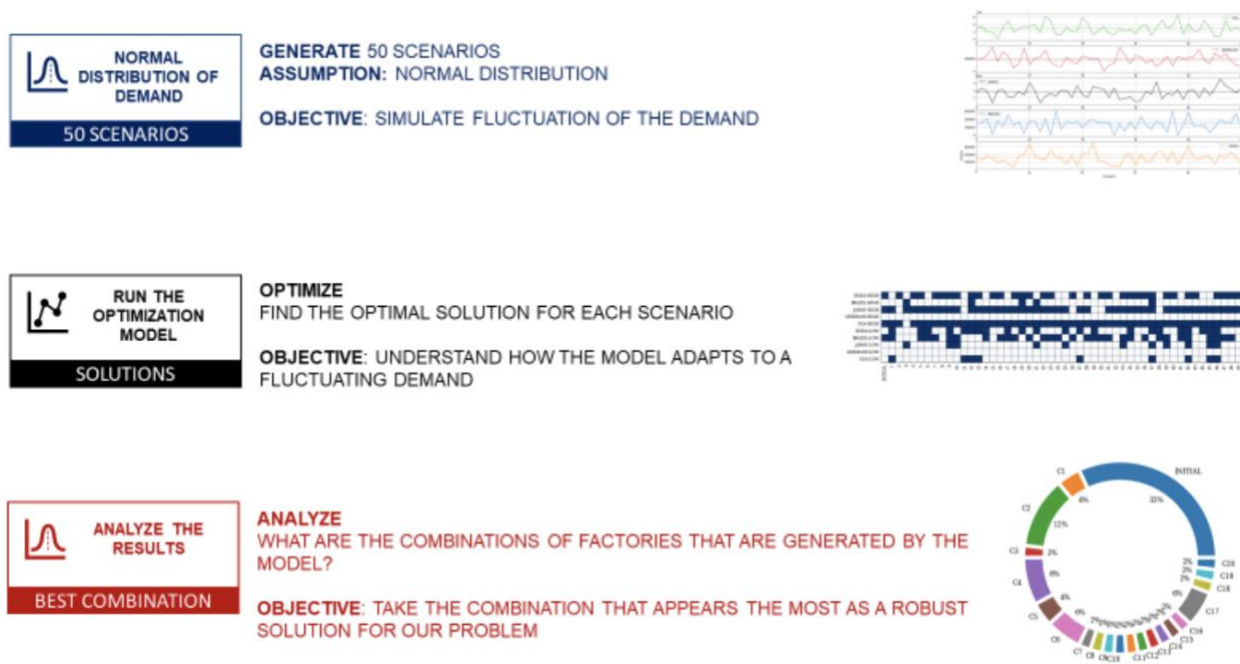


Guardando i tassi di utilizzazione, è facile intuire che la soluzione iniziale non riuscirà ad adattarsi all'incremento della domanda.

## 5. Metodologia di Simulazione

Non si può fare affidamento su un'unica soluzione e, quindi, attendere che la rete assorbirà la domanda durante tutto l'anno.

Solitamente, questi parametri sono calcolati sulla base della media annuale delle vendite.



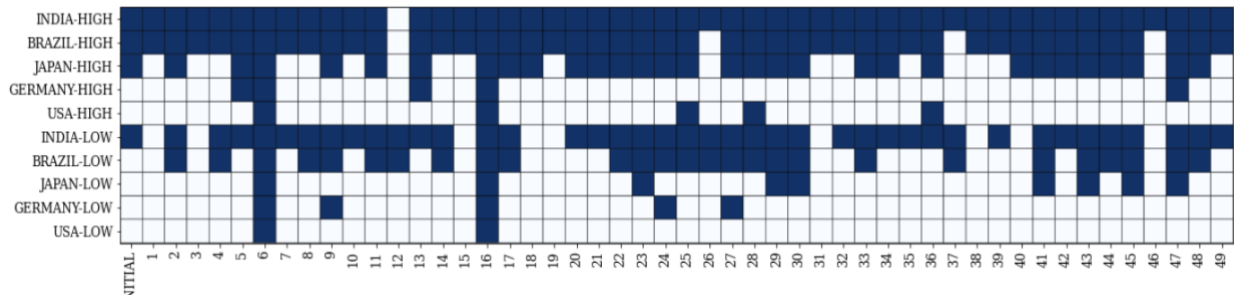
La figura di cui sopra mostra la metodologia di simulazione che si va ad applicare.

In sostanza, si vanno a valutare vari cambiamenti nella domanda, modificandone la variabilità:

- 1) Si generano 50 scenari, assumendo una distribuzione normale della domanda, per simulare la variabilità della stessa;
- 2) Si effettua il run del modello di ottimizzazione, trovando la soluzione ottima per ciascuno dei 50 scenari, con l'obiettivo di capire come il modello si adatta alla variabilità della domanda;
- 3) Si analizzano i risultati, in modo da definire quale soluzione è più robusta per il problema.

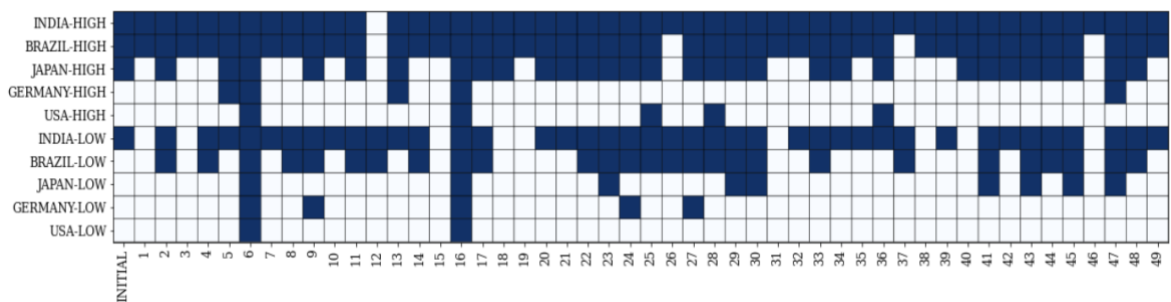
### 5.1. Generate 50 scenarios

Si assume che la domanda segua una distribuzione normale con un coefficiente di variazione  $CV = 0,5$ .



Si ottiene una matrice di 50 colonne che rappresenta i 50 differenti scenari.

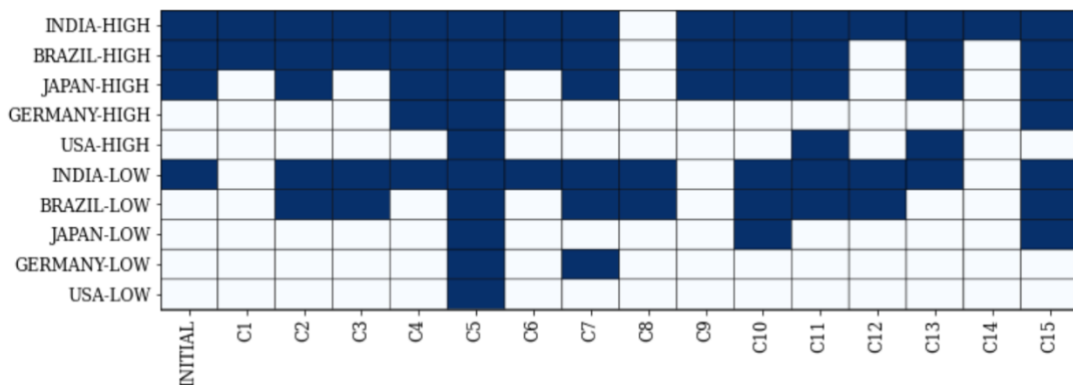
### 5.2. Optimal Solution per ogni scenario



- L'India ha sempre almeno una struttura aperta;
- Lo scenario 12 necessita solo di due strutture a bassa capacità aperte in India e Brasile;
- Lo scenario 16 richiede che tutte le strutture siano aperte causa un picco della domanda in USA.

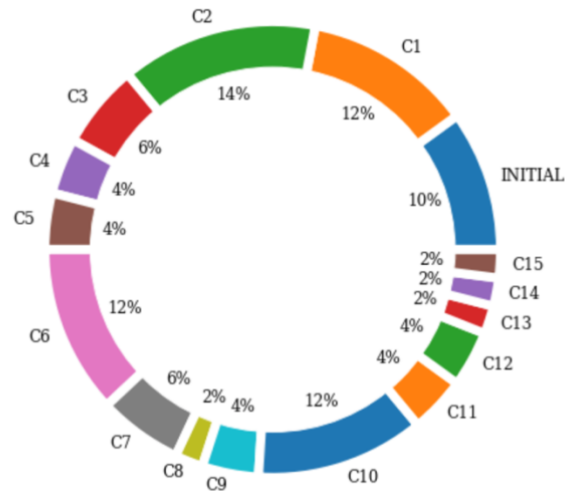
### 5.3. Distribuzione della combinazione ottimale

Si hanno delle combinazioni che appaiono più frequentemente di altre?



Questo sono le combinazioni che si ottengono dopo aver rimosso i duplicati, in numero pari a 16.

Quale di queste combinazioni sono le più robuste?



Sembra che la combinazione C2 (strutture ad alta/bassa capacità in India e Brasile e 1 struttura ad alta capacità in Giappone) sia quella che compare di più. Questa combinazione è seguita da C2, C6 e C10.

Queste combinazioni tendono a massimizzare la produzione nei paesi con bassi costi di produzione e, nella maggior parte degli scenari, riusciamo a soddisfare la domanda.

## 6. *Sviluppi futuri*

Questa metodologia ha creato una serie di candidati come potenziale migliore soluzione, però bisogna testare sui 50 scenari e si dovrà valutare:

- Quale sia il costo totale per ogni scenario;
- Quale sia la percentuale della domanda prodotta dalle strutture.

Questo garantirà maggiori informazioni quantitative.