



UNIVERSITÀ DI PISA

**DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA DEI SISTEMI,  
DEL TERRITORIO E DELLE COSTRUZIONI**

**RELAZIONE PER IL CONSEGUIMENTO DELLA  
LAUREA MAGISTRALE IN INGEGNERIA GESTIONALE**

***Sviluppo di un Algoritmo Genetico per il  
Bilanciamento Multi obiettivo di una Linea di  
Montaggio manuale***

---

RELATORI

Prof. Ing. Gino Dini

*Dipartimento di Ingegneria  
Civile e Industriale*

Ing. Michela Dalle Mura

*Dipartimento di Ingegneria  
Civile e Industriale*

IL CANDIDATO

Claudia Tozzi

claudia.tozzi89@gmail.com

Sessione di Laurea del 25/11/2015  
Anno Accademico 2014/2015  
Consultazione consentita

*Ai miei genitori*



## **Sommario**

Il presente lavoro di tesi si centra sul problema del bilanciamento delle linee di montaggio, con particolare attenzione ad una linea di assemblaggio manuale. Uno degli strumenti che da anni è utilizzato per la risoluzione di tale problema è l'algoritmo genetico, metodo iterativo in grado di individuare soluzioni ottime in tempi di calcolo ragionevoli.

Si è quindi implementato un algoritmo genetico che tiene conto di più obiettivi, dettati e giustificati dalla realtà dei sistemi produttivi e dalle necessità che si ritengono per essi essenziali. Si è utilizzato l'ambiente Matlab sia per implementare l'algoritmo, sia per realizzare delle interfacce grafiche di ausilio nell'inserimento dei dati in input e per la visualizzazione dei risultati ottenuti.

Si è applicato l'algoritmo a due casi di studio e si è dimostrata la sua bontà rispetto ai tradizionali metodi euristici ed iterativi in termini di risultati raggiunti.

## **Abstract**

The following thesis is focused on the assembly line balancing problem, referring in particular way to a manual assembly line. One of the most used search in this type of problem is the genetic algorithm: iterative method able to find the best solutions in not too long times.

We have implemented a multi-objective genetic algorithm, based on the essential characteristics of a real assembly line.

We have used the software "Matlab" both for implement the algorithm that to create the graphic interfaces. The interfaces allow the user to insert data in input and to see the results in output.

We have applied the algorithm at two case studies, seeing his better work, in terms of results obtained, than the usual methods

## Indice

Capitolo 1 Introduzione.....	8
1.1 Contesto generale e motivazione.....	8
1.2 Obiettivi del lavoro di tesi e metodologia impiegata .....	9
1.3 Terminologia impiegata .....	10
1.4 Notazione impiegata.....	11
Capitolo 2 Il montaggio.....	12
2.1 Tipologie di montaggio .....	13
2.2 Linee di montaggio: caratteristiche generali e classificazione .....	15
2.2.1 Numero di prodotti .....	16
2.2.2 Sistemi di movimentazione .....	17
2.2.3 Variabilità dei tempi delle operazioni .....	18
2.2.4 Vincoli durante l'assegnazione delle operazioni.....	19
2.2.5 Layout del sistema di produzione.....	20
2.2.6 Tipologia di stazione .....	22
2.2.7 Strategia di lancio in produzione.....	22
2.3 Analisi di una linea di montaggio manuale .....	23
2.4 Bilanciamento di una linea di montaggio manuale .....	26
2.4.1 Esempio.....	28
2.5 Ottimizzazione del montaggio .....	30
2.5.1 SALBP (Simple assembly line balancing problem).....	33
2.6 Metodi per la risoluzione dei problemi SALBP.....	34
2.6.1 Metodi euristici .....	35
2.6.1.1 Largest candidate rule (LCR).....	35
2.6.1.2 Kilbridge e Wester (K&W).....	36
2.6.1.3 Ranked positional weight (RPW).....	38
2.6.2 Metodi iterativi.....	40
2.6.2.1 Tabu Search.....	40
2.6.2.2 Simulated Annealing .....	40
2.6.2.3 Algoritmo genetico.....	42
Capitolo 3 Algoritmo genetico per la risoluzione di SALBP.....	43
Terminologia generale di un algoritmo genetico .....	43
3.1 Cosa è un algoritmo genetico e come viene utilizzato per risolvere i problemi di montaggio .....	43

3.2 Analisi dello stato dell'arte .....	45
3.2.1 Rappresentazione genetica e generazione della popolazione iniziale di soluzioni.....	45
3.2.2 Valutazione della funzione di fitness per le soluzioni che appartengono all'attuale popolazione .....	47
3.2.3 Processo di riproduzione attraverso la selezione e gli operatori genetici di crossover e mutazione .....	48
3.2.4 Riparazione dei cromosomi non fattibili ed inserimento dei nuovi cromosomi nella popolazione .....	51
3.2.5 Selezione e procedura finale .....	52
3.2.6 Fitness Function .....	53
3.3.1 Risultato delle analisi .....	68
3.4 Considerazioni conclusive sull'analisi dello stato dell'arte .....	69
Capitolo 4 Motivazioni e obiettivi della tesi .....	71
4.1 Obiettivi proposti per valutare la funzione dell'algoritmo genetico .....	71
4.1.1 Minimizzazione del numero di stazioni di lavoro .....	72
4.1.1 Minimizzazione delle abilità degli operatori fra le stazioni .....	72
4.1.2 Minimizzazione degli equipaggiamenti fra le stazioni.....	75
4.1.3 Minimizzazione dei cambi di orientamento dei componenti in una certa sequenza .....	78
4.1.4 Massimizzazione del livellamento del carico di lavoro fra le stazioni.....	80
4.2 Caratteristiche dell'algoritmo genetico implementato .....	81
Capitolo 5 Implementazione e validazione .....	89
5.1 Interfacce grafiche.....	90
5.1.1 Interfaccia grafica iniziale .....	90
5.1.2 Interfaccia grafica del prodotto .....	91
5.1.3 Interfaccia grafica delle operazioni elementari .....	92
5.1.4 Interfaccia grafica dell'algoritmo .....	93
5.1.5 Interfaccia grafica di sintesi dei dati.....	95
5.1.6 Interfaccia grafica per la visualizzazione della miglior soluzione .....	97
5.1.7 Interfaccia grafica per il dettaglio delle informazioni di ciascuna stazione di lavoro .....	99
5.2 Validazione .....	101
5.2.1 Validazione dei limiti inferiori e superiori .....	102
5.2.2 Validazione dei pesi da attribuire alla fitness.....	106
Capitolo 6 Applicazione: Casi di studio.....	108
6.1 Caso di studio: Carburatore Keihin 19 PD.....	108

6.1.1 Descrizione.....	108
6.1.2 Implementazione in “GenIAL” .....	111
6.1.3 Risultati .....	114
6.2 Caso di studio: Tonge.....	124
6.2.1 Descrizione.....	125
6.2.2 Implementazione in “GenIAL” .....	126
6.2.3 Risultati .....	128
6.3 Conclusioni .....	134
Appendice A.....	136
Bibliografia .....	158

## Capitolo 1 Introduzione

### 1.1 Contesto generale e motivazione

Nella moderna produzione industriale risulta sempre maggiore l'esigenza di analizzare i problemi che possono insorgere durante il processo produttivo, ed eliminarne le cause il prima possibile lungo la catena di produzione ha generato un diffuso interesse verso l'analisi dei requisiti del prodotto che nello specifico riguardano il montaggio. Il soddisfacimento di tali requisiti permette una corretta esecuzione dell'operazione di montaggio e perciò la riuscita del prodotto. Sono infatti molte le aziende che ancora oggi operano con sistemi di produzione la cui competitività dipende in gran parte dalla fase di assemblaggio.

L'attività di assemblaggio è un'attività che crea valore, strategica, dalla quale possono dipendere sopravvivenza e successo delle aziende; essa rappresenta una fase critica, nella quale si incontrano i risultati delle fasi di progettazione, ingegnerizzazione e fabbricazione del prodotto.

L'assemblaggio racchiude in sé la funzionalità di tutto il processo produttivo, in quanto punto di incontro di più fasi ed ultima operazione eseguita prima che il prodotto sia pronto per il consumatore.

È inoltre necessario fare le seguenti considerazioni:

- le operazioni di assemblaggio sono tipicamente fra le più costose;
- gran parte degli errori che nascono nelle fasi precedenti della catena si ripercuotono sull'assemblaggio;
- la qualità del prodotto finito dipende fortemente dalla fase di montaggio che, essendo una attività finale, contiene al suo interno anche operazioni quali la messa a punto e il controllo del prodotto, che definiscono a loro volta la qualità dell'intera produzione.

Dato che quindi l'assemblaggio rappresenta una fase così cruciale del processo produttivo, diventa inevitabile per le aziende la ricerca della gestione totale e dettagliata di ogni minimo aspetto che lo riguarda.

Il presente lavoro di tesi si centra sul problema del bilanciamento delle linee di produzione, con particolare riferimento alle linee di assemblaggio manuali. I risultati di un corretto bilanciamento si misurano in termini di minimizzazione di perdite, di costi e di tempi quindi, in fase di progettazione di una linea di assemblaggio, il problema decisionale di come bilanciarla rappresenta una fase critica, la cui gestione riguarda più aspetti (per esempio efficienza, affidabilità, spazi), e la cui esatta riuscita richiede l'impiego di molte risorse.

## **1.2 Obiettivi del lavoro di tesi e metodologia impiegata**

Sotto queste premesse si sviluppa questa tesi che nella prima parte cerca di definire lo stato dell'arte e gli elementi che costituiscono, determinano e influenzano i sistemi di assemblaggio, con particolare riferimento alle linee di produzione manuali.

Dopodiché, con il fine di provare ad individuare la miglior soluzione alle problematiche legate all'attività di montaggio, si introduce uno dei tanti strumenti in grado di risolvere i problemi di ottimizzazione: l'algoritmo genetico. Questo ha ricevuto negli ultimi anni molta attenzione da parte dei ricercatori, in quanto considerato metodo in grado di trovare soluzioni ottime in tempi ragionevolmente brevi.

Si passa perciò alla parte centrale del lavoro nella quale si implementa un algoritmo genetico tenendo conto di obiettivi che, sulla base di valide giustificazioni, si ritengono fondamentali per una gestione più completa possibile di tutti gli aspetti che caratterizzano una linea di montaggio manuale. Il tutto nell'ottica di rendere economicamente più efficiente l'intero impianto. Si applica quindi tale algoritmo a diversi prodotti utilizzati come casi di studio e se ne variano i parametri per giudicarne il comportamento.

L'ultima parte della tesi discute i risultati raggiunti sulla base degli obiettivi proposti e trae le conclusioni.

L'implementazione dello strumento è stata realizzata in ambiente MATLAB, programmando prima l'esecuzione dell'intero algoritmo, e poi una serie di interfacce

grafiche che facilitano in input l'inserimento di dati sia del prodotto che dell'algoritmo, e in output la visualizzazione dei risultati ottenuti.

### 1.3 Terminologia impiegata

Per maggiore chiarezza si introduce la terminologia principale e la notazione che verranno utilizzate in particolare nella prima parte del testo dedicata alla descrizione delle linee di montaggio.

Operazione elementare: operazione di montaggio non ulteriormente scomponibile in attività eseguibili separatamente;

Stazione di lavoro: porzione della linea di montaggio nella quale si svolge una parte del lavoro. Gli elementi identificativi di una stazione sono le gli utensili e gli operatori;

Tempo ciclo: intervallo di tempo che passa dal completamento del montaggio di un oggetto al successivo, ovvero il tempo che ciascuna stazione ha per completare le operazioni di assemblaggio;

Tempo di riposizionamento: tempo necessario in ogni stazione per permettere all'operatore di riposizionarsi o all'unità di movimentazione di spostarsi;

Tempo della stazione: tempo necessario all'operatore della stazione *i-esima* per completare l'operazione di montaggio;

Tempo di attesa: tempo di attesa che si presenta in una stazione nel momento in cui il tempo della stessa è inferiore alla differenza fra il tempo ciclo e il tempo di riposizionamento;

Precedenze: vincoli tecnologici che definiscono l'ordine per il quale le operazioni devono essere assegnate alle stazioni di lavoro;

Bilanciamento: processo di assegnazione delle operazioni elementari alle singole stazioni nel rispetto di tempo ciclo e delle precedenze tra le varie operazioni;

Grafo delle precedenze: diagramma costruito da un set di nodi e uno di archi. I primi rappresentano le operazioni, i secondi le relazioni dirette e quindi le precedenze fra le operazioni stesse;

Capacità della linea: tempo totale disponibile per assemblare un prodotto sulla linea;

Workload smoothness: indice che definisce la distribuzione del carico di lavoro fra le stazioni;

Workload variance: indice che definisce la varianza del carico di lavoro fra le stazioni.

#### 1.4 Notazione impiegata

$N$  = numero di stazioni di lavoro, indice  $i$ ;

$T_c$  = tempo ciclo;

$R_p$  = rateo produttivo;

$T_r$  = tempo di riposizionamento;

$T_{si}$  = tempo necessario alla stazione  $i$  – esima per completare le operazioni;

$T_{max}$  =  $\max\{T_{si}\}$ ;

$T_{ai}$  = tempo di attesa della stazione  $i$ -esima;

$T_{am}$  = tempi di attesa medi lungo la linea;

$T_p$  = tempo necessario per assemblare il prodotto;

$n_e$  = numero di operazioni elementari per completare un dato prodotto, indice  $k$ ;

$T_{ek}$  = tempo necessario per eseguire l'operazione elementare  $k$  – esima;

$C$  = capacità della linea;

$WV$  = Workload Variance

## Capitolo 2 Il montaggio

Il montaggio di un insieme di componenti, effettuato allo scopo di realizzare un prodotto (sia esso un complessivo, un gruppo o un sottogruppo meccanico, elettronico o di altra natura) rappresenta una importante fase del processo che conduce dalla progettazione fino al collaudo del prodotto.

Le operazioni da eseguire durante la fase di montaggio possono essere molteplici, di svariata natura, e possono includere fra l'altro controlli intermedi e prove funzionali del prodotto durante le varie fasi del processo. Le operazioni che vengono eseguite più spesso sono le seguenti:

- manipolazione di oggetti: operazioni preliminari che dipendono dal tipo di montaggio e possono comprendere la presa di un oggetto, il suo spostamento, la sua orientazione, il suo rilascio ecc.;
- accoppiamento tra due componenti: consiste nell'avvicinare un oggetto fino ad alloggiarlo nella sede prevista nella restante parte del complessivo. Nella maggior parte dei casi questa operazione viene eseguita lungo una traiettoria verticale verso il basso, proprio per sfruttare la forza di gravità. Tale operazione è tanto più difficile quanto più è ridotto il gioco fra le superfici o quanto più uno dei due oggetti si trova in un punto difficilmente accessibile;
- avvitatura: operazione che richiede l'utilizzo di appositi strumenti ed utensili (avvitatori) azionati usualmente tramite aria compressa;
- pressatura: anche questa operazione richiede apparecchiature dedicate, costituite da presse con le quali applicare i necessari carichi per poter effettuare unioni fra i componenti che presentino accoppiamenti con interferenza;
- incollaggio: operazione che richiede tre fasi: una prima di preparazione delle superfici e una seconda di deposizione del collante e una terza di unione e pressatura dei componenti;
- deposizione di sigillanti: operazione da utilizzarsi in sostituzione del montaggio di guarnizioni in carta o in gomma;

- controlli intermedi: consistono generalmente in prove funzionali atte a verificare la qualità dei componenti e degli accoppiamenti realizzati fino a quel momento; le prove più comuni sono: verifiche dei giochi, verifiche dei cinematismi, verifiche di continuità elettrica tra i componenti, verifiche di tenute delle guarnizioni, verifiche di rumorosità ecc.

A queste si devono aggiungere tutte quelle operazioni necessarie ad alimentare continuamente la zona di lavoro con i componenti da montare (1).

## 2.1 Tipologie di montaggio

Una delle scelte fondamentali che influenza la convenienza economica di una soluzione piuttosto che di un'altra è la scelta della tipologia di assemblaggio ovvero la scelta fra il montaggio manuale e il montaggio automatico.

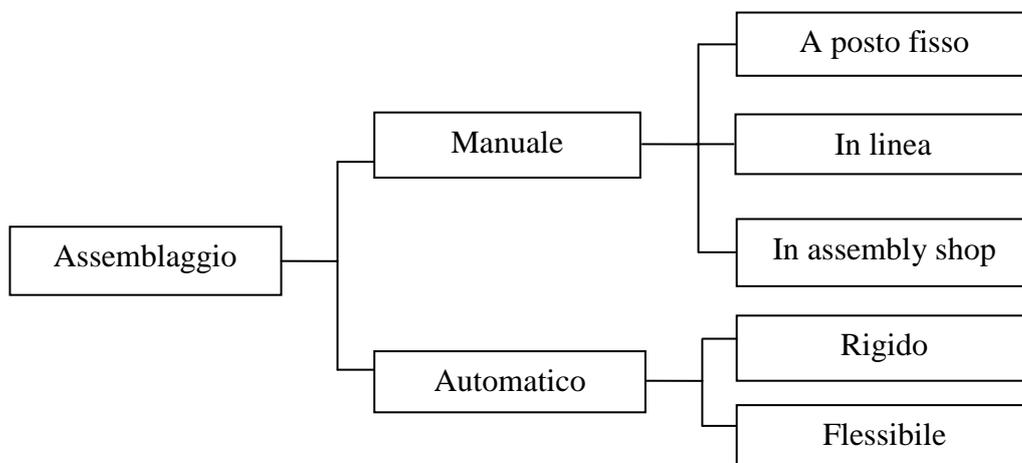


Figura 2.1: Tipologie di montaggio

Il montaggio **manuale** prevede la presenza di operatori che, con l'ausilio di determinati utensili e attrezzature, svolgono direttamente le operazioni necessarie per assemblare un prodotto. Un impianto di montaggio manuale può presentare una delle tre configurazioni mostrate in Figura 2.1:

- impianto di montaggio a **posto fisso**: tutte le attività sono svolte in un'unica postazione sulla quale operano una o più persone, quindi il prodotto da realizzare non si trasferisce da una postazione di montaggio ad una successiva. Tale configurazione presenta tutte le parti e le attrezzature necessarie nella medesima postazione ed è particolarmente adatta al montaggio di grossi prodotti quali, per esempio, grandi turbine, aeroplani, vagoni ferroviari ecc. E' dunque una soluzione da preferirsi per produzioni di medio-bassi volumi e dalla necessità di limitare i movimenti di componenti pesanti ed ingombranti.
- impianto di montaggio **in linea** (linea o catena di montaggio manuale): costituito da una linea di produzione realizzata da una sequenza di stazioni di lavoro dove le operazioni di montaggio sono eseguite da operatori; ciascuno di essi svolge una parte del montaggio. Ciascuna stazione, nella quale possono essere presenti uno o più operatori, è alimentata con i particolari da montare e con i dispositivi necessari a compiere le operazioni di montaggio. Tipicamente il montaggio parte da un componente di base che viene immessa all'inizio della linea, e così, passando da una stazione alla successiva, essa accoglie tutti gli altri componenti fino ad arrivare al termine della linea completamente montato.
- impianto di montaggio **assembly shop**: questa configurazione non presenta un collegamento seriale fra le stazioni. Queste non sono disposte secondo una linea e i collegamenti sono più flessibili in funzione del particolare ciclo di montaggio. L'area di lavoro è suddivisa in celle all'interno delle quali si svolgono specifiche operazioni. Il prodotto passa da una cella all'altra in funzione della sequenza di operazioni previste dal suo ciclo di montaggio.

Il montaggio **automatico** prevede l'utilizzo di robot e altre attrezzature che permettono di sostituire l'uomo in particolar modo per le operazioni più semplici e ripetitive nelle quali è richiesta minore abilità da parte dell'operatore. Questo perché maggiore è la complessità della operazione, maggiore è il costo del dispositivo automatico che effettua tale operazione.

In generale si può affermare che l'automazione è tanto più economicamente vantaggiosa quanto più sarà grande la serie da produrre; infatti per piccole dimensioni del lotto è

conveniente il montaggio manuale mentre per grandi dimensioni quello automatico. Quest'ultimo può a sua volta essere:

- montaggio automatico **rigido**: effettuato con macchine dedicate ad un singolo prodotto;
- montaggio automatico **flessibile**: impiega macchine dotate di elevate flessibilità che consentono sulla stessa linea il montaggio di diverse varianti del prodotto.

La tipologia di montaggio cui si fa riferimento in questa tesi è quella manuale con configurazione in linea. Di seguito si riportano le caratteristiche generali delle linee di montaggio e la loro classificazione.

## 2.2 Linee di montaggio: caratteristiche generali e classificazione

Le linee di montaggio sono sistemi tipici nella produzione industriale caratterizzati da elevate quantità di prodotti. Come già accennato nel precedente paragrafo, esse fanno parte di quelle configurazioni con disposizione sequenziale di stazioni di lavoro, tale da rispecchiare la sequenza tecnologica di operazioni necessarie al completamento del prodotto. Per questo motivo in letteratura vengono indicate come sistemi produttivi *process-oriented*, in contrapposizione a quelli *job-oriented*, nei quali invece sono raggruppate per funzione.

La disposizione più semplice consiste in un certo numero di stazioni collegate fra loro da un sistema di trasporto di materiali (tipicamente un nastro trasportatore); i semilavorati vengono caricati lungo la linea e trasportati da una stazione all'altra in ognuna delle quali viene svolta una frazione del lavoro necessario al completamento del prodotto.

Nella letteratura generalmente le linee di montaggio sono classificate rispetto ad alcuni fattori critici che le caratterizzano, i quali devono essere fissati e gestiti durante la progettazione.

Di seguito si espone la classificazione fornita da Scholl (2).

### 2.2.1 Numero di prodotti

Si distinguono tre tipologie di linee in base al numero di prodotti assemblati:

- linee **single-model**: prevedono che un unico prodotto sia realizzato in elevate quantità lungo la linea;



Figura 2.2: Schematizzazione linea single-model

tutte le stazioni ripetono le stesse operazioni su identici semilavorati, quindi il carico di lavoro delle stazioni rimane costante;

- linee **mixed-model**: prevedono che molte versioni del prodotto di base siano montate sulla stessa linea. I processi che interessano le diverse versioni sono simili in quanto i prodotti differiscono solamente per alcuni attributi e non per caratteristiche essenziali. Ci sono perciò delle operazioni principali che si ripetono in ogni modello più un insieme di operazioni che si distinguono per il tempo di esecuzione o che possono non esserci, a seconda della specifica versione;



Figura 2.3: Schematizzazione linea mixed-model

- linee **multi-model**: prevedono la produzione di diversi prodotti lungo la stessa linea. In questo caso vi possono essere differenze sostanziali nel processo produttivo per cui è necessario un set-up della linea ogni volta che si passa da un prodotto ad un altro. Quindi, per minimizzare i costi di set-up, la produzione avviene per lotti.



Figura 2.4: Schematizzazione linea multi-model

## 2.2.2 Sistemi di movimentazione

Il sistema di movimentazione dei componenti che costituiscono un prodotto deve svolgere le seguenti funzioni: alimentare l'impianto con i componenti che provengono dall'esterno, allontanare dall'impianto i prodotti assemblati e movimentare il prodotto all'interno dell'impianto lungo le varie fasi del montaggio (1). Tali funzioni si possono svolgere con sistemi manuali o automatici.

### Sistemi di movimentazione manuali

Con questi sistemi il prodotto viene movimentato da una stazione all'altra attraverso gli operatori stessi i quali, una volta terminata l'operazione, spostano il prodotto alla stazione successiva. Con questo sistema si possono presentare due tipi di problemi:

- situazione di **starving**: l'operatore termina l'operazione sul prodotto, lo passa alla stazione successiva, ma il prossimo prodotto non arriva quindi l'operatore rimane in posizione di attesa;
- situazione di **blocking**: l'operatore termina l'operazione sul prodotto, lo passa all'operatore della stazione successiva che però non lo può ricevere in quanto non ha ancora terminato il suo lavoro.

Questi inconvenienti possono essere limitati o eliminati attraverso l'utilizzo di *buffer* (magazzini intermedi) che sono in grado di ricevere un certo numero di unità.

### Sistemi di movimentazione automatici

In questo caso ci si riferisce a sistemi automatici a rotaia o a nastro i quali consentono al prodotto di passare in modo automatico e sequenziale tra le varie stazioni di montaggio. Tipicamente il sistema è costituito da un insieme di pallet che scorrono su un nastro convogliatore; ciascun pallet trasporta il prodotto e lo presenta correttamente presso ogni stazione. I metodi di movimentazione automatica possono essere divisi in tre categorie fondamentali:

- **continui:** effettuano il movimento del prodotto a velocità costante;
- **discontinui sincroni:** le unità sono mosse da una stazione all'altra simultaneamente, tramite un moto discontinuo;
- **discontinuo asincrono:** una unità si muove dalla stazione di lavoro per andare alla successiva non appena l'operatore termina il proprio compito e dà il consenso per il movimento.

In riferimento al tempo di sosta in una stazione, ovvero il tempo massimo nel quale un semilavorato può sostare in ciascuna stazione, questo dipende da quale categoria fra le precedenti descritte si sta considerando.

Sia i sistemi continui che i discontinui sincroni sono regolati da una cadenza fissa che è determinata dal tempo di montaggio della stazione più lenta; nei sistemi asincroni invece non esiste alcuna cadenza determinata a priori, ma l'operatore rilascia l'unità non appena ha completato l'operazione.

### 2.2.3 Variabilità dei tempi delle operazioni

A seconda del tipo di operazione e dell'operatore, i tempi delle operazioni possono più o meno variare ed essere soggette ad un grado di incertezza generato da varie cause. In particolare, nelle linee di montaggio manuali, i tempi delle operazioni sono soggetti ai fattori fisici, psicologici e sociali degli operatori.

In linea generale si può assumere che se le operazioni sono semplici, i tempi saranno soggetti a minore incertezza; viceversa, se le operazioni sono complesse, essi saranno soggetti ad una maggiore incertezza in quanto si può incorrere in un numero maggiore di eventi aleatori.

Si possono individuare tre diverse modalità con cui si possono assumere i tempi delle operazioni:

- tempi delle operazioni **deterministici:** essi si riferiscono a tempi fissi delle operazioni. Questa assunzione si giustifica dal momento che la variabilità stimata dei tempi delle operazioni è sufficientemente piccola. Se la linea è manuale, gli operatori devono essere altamente preparati e motivati ed il metodo deve essere

altamente robusto e sofisticato; a crescere invece del grado di automazione è naturale che i tempi siano soggetti ad un decrescente grado di variabilità;

- tempi delle operazioni **stocastici**: nel caso di linee di montaggio manuali, variazioni significative dei tempi delle operazioni possono influenzare fortemente la performance del sistema; questo perché non si possono non prendere in considerazione le abilità e le competenze degli operatori, il loro tasso di lavoro ed altri fattori. Perciò i tempi stocastici tengono conto dei vari inconvenienti nei quali si può incorrere;
- tempi delle operazioni **dinamici**: essi tengono conto del fenomeno di apprendimento che investe gli operatori. Nei periodi successivi alla installazione di una nuova linea è logico pensare che i tempi delle operazioni decrescono dal momento che un operatore acquisisce dimestichezza con le operazioni che deve eseguire. La difficoltà sta però nel quantificare questo fenomeno, che in ogni modo dipende dalla natura delle operazioni, dalla flessibilità del sistema produttivo e dalla velocità di apprendimento degli operatori.

#### 2.2.4 Vincoli durante l'assegnazione delle operazioni

Oltre ai vincoli di precedenza, vi sono anche ulteriori costrizioni che vanno prese in considerazione nell'assegnazione delle operazioni:

- vincoli **legati all'operazione**: si presentano nel momento in cui due o più operazioni devono necessariamente essere assegnate alla stessa stazione o porzione di linea. Tale condizione si verifica per esempio quando le operazioni necessitano della stesso stato (temperatura ecc.). È inoltre possibile che si verifichi la condizione opposta, ovvero che due operazioni non possono essere assegnate alla stessa stazione;
- vincoli **legati alla stazione**: si presentano per le operazioni che si effettuano con un certo macchinario che non è possibile spostare in un'altra stazione a costi accettabili. Restrizione simile si ha per l'approvvigionamento delle materie prime, qualora queste siano disponibili solamente in una data stazione della linea per motivi di approvvigionamento;

- vincoli **legati alla posizione**: legati al fatto che è preferibile effettuare il minor numero di rotazioni possibili al semilavorato, per motivazioni economiche e di fattibilità. Risulta perciò necessario raggruppare le operazioni da effettuare in una certa posizione nella medesima sezione della linea;
- vincoli **legati all'operatore**: sorgono nel momento in cui un'operazione è assegnata ad uno specifico operatore. La qualifica di un operatore è determinata dall'operazione più complessa che è in grado di svolgere; per questo è necessario concentrare operazioni più complesse in poche stazioni.

### 2.2.5 Layout del sistema di produzione

Il layout della linea di montaggio è nella maggior parte dei casi determinato dalla sequenza delle lavorazioni necessarie al corretto fluire dei semilavorati. Ciò nonostante, si distinguono diverse possibili configurazioni:

- **linee in serie**: tradizionale disposizione delle stazioni lungo un linea retta;

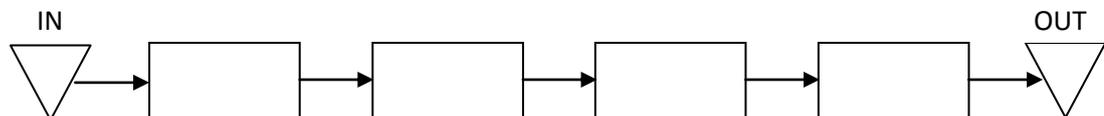


Figura 2.5: Linea in serie

- **linee in parallelo**: installate per incrementare la flessibilità dei sistemi nella realizzazione di uno o più prodotti; presentano numerosi vantaggi fra i quali: maggiore velocità di adattamento alla variazione della domanda del mercato, riduzione dei fermi causa rottura di macchinari (non si blocca tutta la linea, cosa che avverrebbe nelle linee in serie) e maggiore intervallo di tempi ciclo così da permettere un miglior bilanciamento della linea;
- **linee ad U**: configurazione nella quale i macchinari sono disposti lungo una linea ripiegata ad U su se stessa secondo l'ordine delle operazioni necessarie alla realizzazione del prodotto. Questo layout ha la capacità di sfruttare le competenze degli operatori multi-specializzati, i quali possono operare contemporaneamente in più punti della linea. La forma inoltre concede una maggiore comunicazione che significa migliore controllo, più rapido fermo della linea in caso di

malfunzionamenti e necessità di un minor numero di macchinari poiché è possibile raggruppare operazioni simili su due lati;

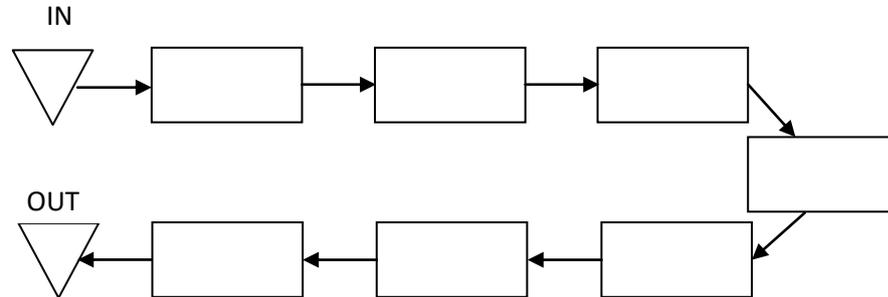


Figura 2.6: Linea ad U

- **stazioni in parallelo:** in questo caso non si replica la linea ma si installano stazioni multiple; i semilavorati si distribuiscono fra più operatori che eseguono le stesse operazioni e, come nel caso di linee in parallelo, è necessaria l'installazione di più macchinari;

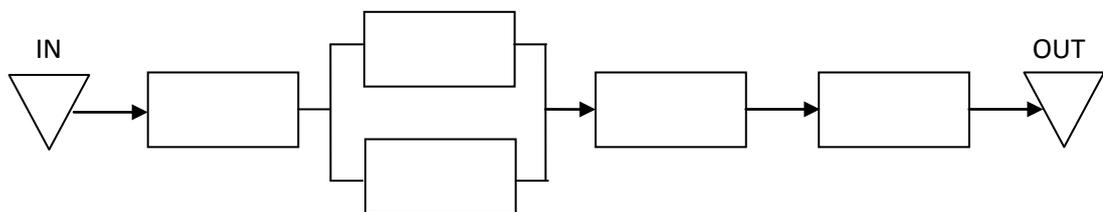


Figura 2.7: Stazioni in parallelo

- **linea a due lati:** linee composte da due normali linee singole disposte in parallelo fra loro; in pratica sul semilavorato operano coppie di stazioni in parallelo sui due lati dello stesso. È infatti una configurazione particolarmente adatta al montaggio di prodotti pesanti, oppure, nel caso in cui il prodotto evidenzi simmetrie di lavorazione, anche a prodotti di dimensioni ridotte;
- **tavola rotante:** usata al posto del nastro trasportatore come sistema di movimentazione, con rotazione permessa grazie ad un motore pneumatico. Gli operatori sono disposti attorno ad un tavolo che ha la capacità di ruotare con un numero fisso di postazioni manuali o automatizzate. Tale configurazione consente elevata ripetitività dei compiti e la possibilità di assegnare più operazioni allo stesso

operatore; l'intero lotto di una famiglia di prodotti è completato in  $n$  giri della tavola (con  $n \geq 2$  altrimenti si riconduce ad una postazione in linea tradizionale).

### 2.2.6 Tipologia di stazione

Le stazioni di lavoro, a seconda del tipo di attrezzature e del sistema di controllo possono presentare diversi gradi di automazione e flessibilità. Si distinguono:

- stazioni **manuali**: le operazioni sono svolte da un operatore umano che utilizza semplici attrezzi o macchine utensili;
- stazioni **semi-automatizzate**: intervento umano limitato alle operazioni di controllo e rifornimento materiale per i macchinari i quali operano automaticamente;
- stazioni **totalmente automatizzate**: qualunque tipo di operazione viene svolta interamente dai macchinari, senza l'ausilio di operatori umani.

Nelle stazioni manuali si fa una ulteriore distinzione tenendo conto della possibilità o no che l'operatore ha di eseguire le operazioni al di fuori della linea, la quale è fisicamente delimitata dalla sua stazione:

- stazioni **chiuse**: accolgono tipicamente operazioni che richiedono particolari condizioni ambientali limitate alla stazione stessa; l'operatore non può oltrepassare i limiti e la lavorazione deve essere completata all'interno della stazione;
- stazioni **aperte**: l'operatore può oltrepassare i confini della stazione ma l'entità del passaggio è limitata per esempio dagli utensili o dal sistema di movimentazione dei materiali.

### 2.2.7 Strategia di lancio in produzione

Si distinguono due tipologie di intervalli di lancio:

- lancio **ad intervalli fissi**: prevede che, nelle linee sincrone, la lavorazione di un nuovo prodotto inizi ad ogni intervallo di tempo corrispondente al tempo ciclo; i

semilavorati risultano così equidistanti fra di loro lungo il sistema di movimentazione;

- lancio **ad intervalli variabili**: soluzione più flessibile ed in grado di minimizzare il tempo “vuoto” nella prima stazione in quanto la seconda unità è lanciata in produzione nel momento in cui la stazione ha completato le operazioni necessarie sulla prima. Dal momento però che i tempi ciclo della stazione sono soggetti a variabilità, i semilavorati non risulteranno equidistanti lungo il sistema di movimentazione e perciò si possono creare situazioni di starving o blocking descritte precedentemente. Tali problemi si possono risolvere cercando di concentrare l’attenzione sulle stazioni collo di bottiglia (quelle nelle quali il tempo ciclo risulta essere maggiore). Si sceglie l’intervallo di lancio quindi saturando per prima queste stazioni, ovvero ponendo il tempo di lancio pari al tempo ciclo di tale stazione.

Si riporta di seguito l’analisi ed il bilanciamento di una linea di montaggio manuale, non solo per semplicità, ma soprattutto perché rappresenta il tipo di linea frutto di studio di questa tesi.

### 2.3 Analisi di una linea di montaggio manuale

Questa analisi considera una linea di montaggio manuale single-model con un operatore per ciascuna stazione  $N$  come rappresentato in Figura 2.8. Si ipotizza che questa sia una linea a cadenza (continua o discontinua sincrona).

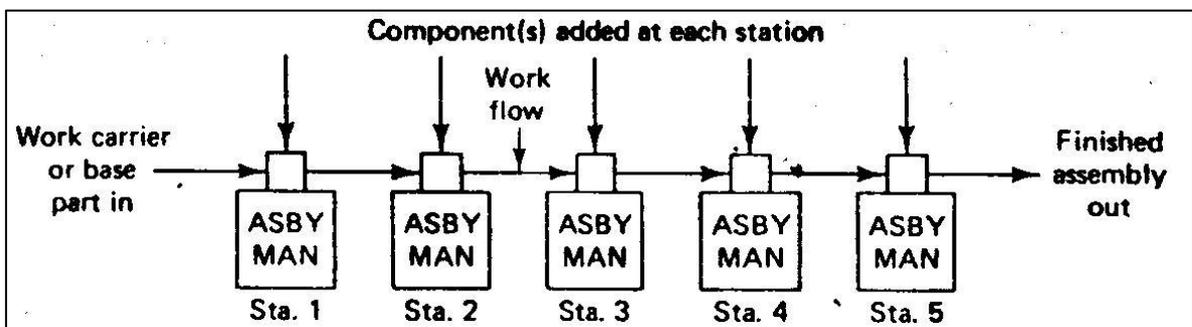


Figura 2.8: Rappresentazione schematica di una linea di montaggio manuale

Quando si dimensiona una linea di montaggio, l'obiettivo è quello di raggiungere un certo rateo produttivo  $R_p$  tale da soddisfare la domanda di un determinato prodotto. La domanda di un prodotto si determina come quantità annua  $D_a$ , la quale deve essere convertita in produzione oraria. A tal fine è necessario conoscere: il numero di settimane per anno  $W_y$ , il numero di turni a settimana  $S_w$  e il numero di ore per turno  $H_s$  nelle quali la linea deve funzionare. Il rateo produttivo è dato quindi da:

$$R_p = D_a / (W_y S_w H_s)$$

A sua volta, il rateo produttivo deve essere trasformato in cadenza, ovvero in un tempo ciclo  $T_c$  che rappresenta l'intervallo di tempo che passa dal completamento dell'assemblaggio di un oggetto al successivo e quindi rappresenta anche il tempo che ciascuna stazione ha per completare le operazioni di montaggio (1). Tale tempo si calcola come:

$$T_c = 60 / R'_p$$

dove  $R'_p$  rappresenta il rateo produttivo che si avrebbe nel caso in cui non fossero presenti perdite produttive dovute all'inefficienza della linea, come mancanza di pezzi, fermi, ecc.

Il rapporto fra  $R_p$  e  $R'_p$  è detto **efficienza della linea**:

$$\eta_L = R_p / R'_p$$

Nelle linee manuali generalmente l'efficienza della linea è compresa fra 0.90 e 0.98.

L'efficienza della linea è un parametro di fondamentale importanza ed è responsabilità del management mantenerlo il più possibile prossimo al 100 % attraverso varie azioni che possono includere per esempio: programma di manutenzione preventivo tale da fermare il numero di fermate per guasto, squadre di intervento ben addestrate allo scopo di ridurre i tempi di fermo, gestione efficiente dei componenti in ingresso e una ottima cura degli approvvigionamenti dall'esterno in modo da evitare i fermi dovuti a scarsa qualità.

Se si sostituisce il valore di tale efficienza nella espressione precedente, si ricava il tempo ciclo in minuti/pezzo pari a:

$$T_c = 60 \eta_L / R_p$$

Questo per evidenziare che un tempo ciclo calcolato come nella prima equazione non permetterebbe di mantenere il rateo produttivo  $R_p$ , ed è perciò necessario utilizzare un tempo ciclo minore.

Durante l'analisi di una linea di montaggio, altro elemento da valutare è il numero di operatori necessari per assemblare un determinato prodotto, ovvero il numero minimo di stazioni di lavoro. Tale elemento si valuta in funzione del rateo produttivo e del tempo totale  $T_p$  necessario per compiere tutte le operazioni di assemblaggio eseguite nelle varie stazioni. Il numero minimo di stazioni è dato da:

$$N = CL/TD$$

dove  $CL$  rappresenta il carico di lavoro da completare in un certo periodo di tempo e  $TD$  il tempo disponibile in quel periodo. Il carico di lavoro si valuta come:

$$CL = R_p T_p$$

Se  $R_p$  è espresso in pezzi/ora e  $T_p$  in min/pezzo, il carico di lavoro è espresso in minuti di lavoro per ogni ora.

Il tempo disponibile è invece il periodo di riferimento moltiplicato per l'efficienza della linea. Nel caso in cui il periodo considerato sia un'ora, il tempo disponibile è valutabile quindi come:

$$TD = 60 \eta_L$$

Un peggioramento della efficienza della linea può provocare un aumento del numero di stazioni necessarie. Il tempo disponibile, tuttavia, non tiene conto di due aspetti molto importanti: le perdite dovute al riposizionamento (in ogni stazione viene perso del tempo per permettere all'operatore di riposizionarsi) e la suddivisione del carico di lavoro (è praticamente impossibile suddividere il carico equamente fra le stazioni).

Quindi si deve tener conto di ulteriori parametri:  $T_r$  e  $T_{si}$ .

La stazione che presenta  $T_{max} = \max\{T_{si}\}$  viene detta stazione "collo di bottiglia". È chiaro che deve risultare:

$$T_{max} \leq T_c - T_r$$

Se dovesse risultare  $T_{max} < T_c - T_r$ , il valore  $T_c$  calcolato precedentemente in funzione della domanda del mercato potrebbe essere ridotto fino ad ottenere:  $T_{max} = T_c - T_r$  e quindi  $T_c = T_{max} + T_r$ , permettendo quindi un aumento del rateo produttivo.

Per tutte le altre stazioni nelle quali  $T_{si} = T_c - T_r$  sarà quindi presente un tempo di attesa, detto “idle time”, dato da:

$$T_{ai} = T_c - (T_{si} + T_r)$$

Le perdite dovute ai riposizionamenti possono essere misurate attraverso l'**efficienza di riposizionamento** della linea, definita come:

$$\eta_r = T_{max} / T_c = (T_c - T_r) / T_c = 1 - T_r / T_c$$

## 2.4 Bilanciamento di una linea di montaggio manuale

Tipicamente le varie stazioni presentano “idle time” diversi e il tempo ciclo si fissa prendendo in considerazione la stazione ‘collo di bottiglia’. Se si volesse ridurre ulteriormente il tempo ciclo, sarebbe necessario cercare di suddividere più equamente il carico di lavoro fra le varie stazioni in modo da tendere alla situazione limite nella quale tutte le stazioni hanno lo stesso  $T_{si}$  e quindi tempi di attesa nulli. Tale problema prende il nome di bilanciamento della linea e ha lo scopo di trovare soluzioni che permettono di aumentare la produttività dell’impianto.

Si introduce il concetto di operazione elementare, ovvero l’operazione di montaggio che non può essere ulteriormente scomposta in operazioni eseguibili separatamente. Si indica con  $T_{ek}$  il tempo necessario per eseguire l’operazione elementare k-esima e con  $n_e$  il numero totale di operazioni elementari per completare un dato prodotto, quindi il tempo necessario per assemblare il prodotto è:

$$T_p = \sum T_{ek} \text{ con } k = 1, \dots, n_e$$

Se si considera un’unica stazione i-esima, il tempo necessario per completare tutte le operazioni elementari che sono ad essa associate è dato da:

$$T_{si} = \sum T_{ek} \text{ con } k = \text{variabile in un sotto intervallo di } 1, \dots, n_e$$

Deve perciò risultare:

$$T_p = \sum T_{si} \text{ con } i = 1, \dots, N$$

L'assegnazione delle varie operazioni elementari alle stazioni dipende da molteplici fattori quali per esempio: similitudini fra operazioni successive oppure legami fra operazioni successive.

I precedenti vincoli, insieme al fatto che i vari  $T_{ek}$  sono differenti fra loro, non permettono di fatto di ottenere un distribuzione bilanciata del carico di lavoro fra le stazioni. Si può però cercare soluzioni migliori cambiando per esempio la sequenza di esecuzione delle operazioni; tuttavia una sequenza di montaggio non può essere modificata senza criterio in quanto si deve tener conto delle precedenze tecnologiche da rispettare.

Si definisce quindi bilanciamento di una linea di montaggio come il processo di assegnazione delle operazioni di montaggio alle singole stazioni nel rispetto di tempo ciclo e precedenze tra le varie operazioni.

Tipicamente si segue la logica di ridurre il più possibile il numero di stazioni lungo la linea, saturando al massimo il tempo possibile, così da ridurre il costo della manodopera.

Questo però non è sempre vero perché in alcuni casi il bilanciamento deve essere fatto tenendo conto dei costi di produzione, dati da:

$$C_p = C_m + C_{nc} \text{ [Euro/pezzo]}$$

dove  $C_m$  è il costo della manodopera e  $C_{nc}$  il costo atteso di non completamento del prodotto, ovvero il costo che si deve sostenere se una o più stazioni non completano le operazioni a loro assegnate.

Se l'obiettivo è quello di saturare sempre di più la linea, aumentando il contenuto di lavoro per ciascuna stazione, si ha una diminuzione di  $C_m$  ma anche un aumento di  $C_{nc}$  in quanto aumenta la probabilità che alcune operazioni non siano completate. Da queste considerazioni si può dedurre che un ulteriore approccio può essere quello di bilanciare la linea cercando di non saturarla completamente, ma di raggiungere una percentuale di saturazione a cui corrisponde il minimo costo di produzione.

In ogni modo, qualunque sia l'approccio, una volta che si raggiunge una determinata soluzione, se ne misura la bontà attraverso l'**efficienza di bilanciamento**. Essa è data da:

$$n_b = T_p / (N T_{\max})$$

Valori tipici dell'industria di tale parametro oscillano fra 0.90 e 0.95.

Si introduce quindi un'ultima grandezza che è data dal prodotto dei tre fattori che misurano l'efficienza dell'impianto, ovvero l'**efficienza globale**:

$$\eta = \eta_l \eta_r \eta_b$$

Utilizzando tale fattore è possibile calcolare un valore più realistico del numero di stazioni sulla linea di montaggio:

$$N = R_p T_p / (60 \eta)$$

### 2.4.1 Esempio

Si illustra in Figura 2.9 un esempio di montaggio di un prodotto, un phon con 17 componenti che devono essere montati secondo una certa sequenza di operazioni di accoppiamento e di avvitatura.

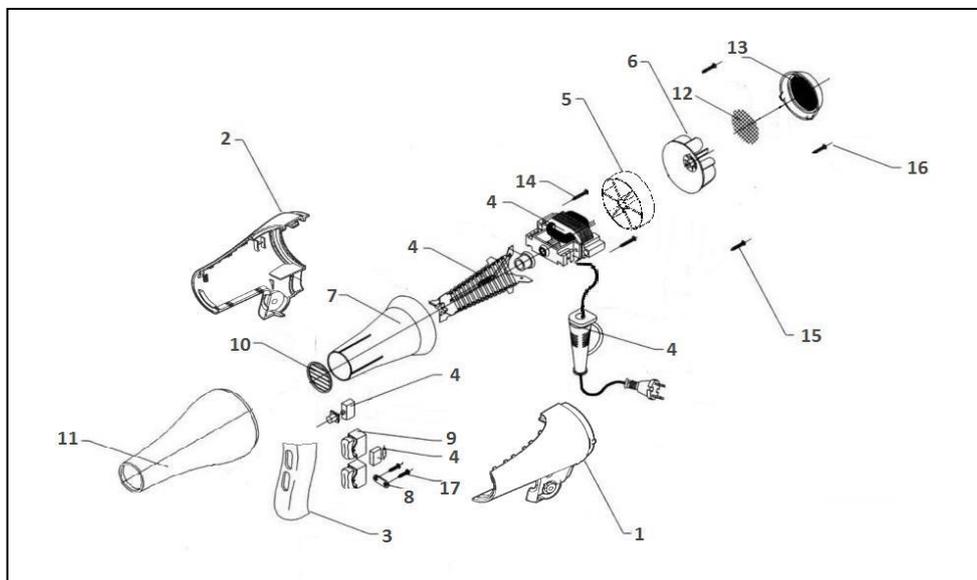


Figura 2.9: Esempio di montaggio di un prodotto: phon

Nella Tabella 2.1 sono riportate le operazioni elementari con i relativi tempi e le precedenze. Si riporta inoltre il grafo delle precedenze in Figura 2.10.

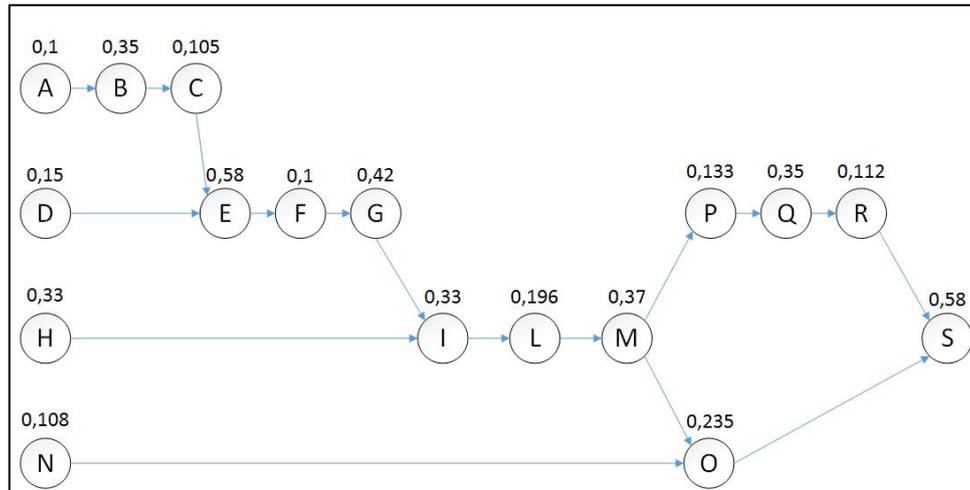


Figura 2.10: Grafo delle precedenze del phon

Op.	Descrizione dell'operazione elementare	$T_{ek}$ [min]	Precedenze
A	Inserire il contenitore n.5 per l'alloggiamento della ventola n.6 nel componente principale n.4	0,1	-
B	Avvitare il contenitore nel componente principale con due viti n.14	0,35	A
C	Bloccare la ventola n.6 nell'albero del n.4	0,105	B
D	Inserire il cono di rivestimento n.7 nel resistore del n.4	0,15	-
E	Posizionare n.4+n.5+n.6+n.7 nel contenitore n.1	0,58	C,D
F	Bloccare n.4 nel n.1 con la piastra n.8	0,1	E
G	Avvitare con due viti n.17	0,42	F
H	Posizionare i bottoni n.9 nel contenitore frontale n.3	0,33	-
I	Bloccare n.3 con n.1 e n.9 con il circuito elettrico del n.4	0,33	G,H
L	Posizionamento della copertura n.2	0,196	I
M	Avvitare con due viti n.15	0,37	L
N	Bloccare la griglia n.10 nel cono frontale n.11	0,108	-
O	Bloccare n.10+n.11 con n.1+n.2	0,235	M,N
P	Posizionare la griglia n.12 on n.1+n.2	0,133	M
Q	Avvitare con due viti n.16	0,35	P
R	Bloccare il filtro n.13 con n.12	0,112	Q
S	Test finale	0,58	O,R

Tabella 2.1: Elenco delle operazioni elementari relative al montaggio del prodotto illustrato in Figura 2.3

La linea deve essere dimensionata e bilanciata secondo una domanda annua di 160.000 phon; inoltre la linea lavora per 50 settimane all'anno, 5 turni a settimana e 8 ore per turno.

Si ipotizza un unico operatore per stazione, una efficienza di linea del 95 % e un tempo di riposizionamento di 0.05 min. Si calcola quindi per primo il rateo produttivo:

$$R_p = 160.000 / (50 * 5 * 8) = 80 \text{ phon/ora.}$$

Il tempo ciclo corrispondente è pari a:  $T_c = 60 * 0.95 / 80 = 0,71 \text{ min}$

e il tempo necessario ad assemblare completamente il prodotto è:  $T_p = 4,549 \text{ min.}$

Il numero minimo teorico di operatori e quindi di stazioni di lavoro si ricava come:

$$N = 4,549 / 0,71 = 6,4 \text{ quindi } N = 7.$$

Il tempo disponibile per eseguire il montaggio è dato da:  $T_{max} = 0,71 - 0,05 = 0,66 \text{ min}$  e l'efficienza di riposizionamento della linea è così data da:  $\eta_r = 0,66 / 0,71 = 92,9 \%$ .

Di conseguenza si ricava che l'efficienza di bilanciamento è data teoricamente da:

$$\eta_b = 4,549 / (6 * 0,66) = 114 \%;$$

essendo tale valore superiore al 100 %, necessariamente il numero minimo di operatori deve essere aumentato di un'unità. Risulta quindi  $N = 8$  e quindi:

$$\eta_b = 4,549 / 7 * 0,66 = 0.98 \%.$$

Il fattore di efficienza globale della linea è dato perciò da:  $\eta = 0.95 * 0.929 * 0.98 = 0.86 \%$

Utilizzando tale valore è possibile ricalcolare un valore più realistico del numero di stazioni necessarie per la linea di assemblaggio:

$$N = (80 * 4,549) / (60 * 0,86) = 7,05 = 8.$$

## 2.5 Ottimizzazione del montaggio

Sulla base di quanto descritto fino ad ora si ritiene necessario evidenziare quali possono essere i punti chiave su cui agire per raggiungere soluzioni ottime di assemblaggio.

Due sono gli aspetti principali di cui tener conto per l'ottimizzazione del montaggio.

Il primo aspetto riguarda la generazione di una specifica sequenza di montaggio di un prodotto: **ASP** (Assembly sequence planning). In particolare esso ha lo scopo di definire la sequenza di operazioni che si possono effettuare per assemblare un gruppo di elementi; tali operazioni possono essere limitate dalle proprietà geometriche e meccaniche degli elementi e dalla loro configurazione. I principali obiettivi di questo primo aspetto sono:

- minimizzazione del numero di cambiamenti nelle direzioni di assemblaggio;
- minimizzazione del numero di cambiamenti di utensili;
- minimizzazione dei cambiamenti nella tipologia di operazione elementare;
- minimizzazione dei costi;
- minimizzazione dei tempi;
- minimizzazione della distanza da percorrere per gli utensili.

Il secondo aspetto, già in parte discusso, riguarda il bilanciamento della linea di montaggio di un prodotto: **ALB** (Assembly line balancing); esso si riferisce al bilanciamento di un certo set di operazioni ognuna delle quali ha un certo tempo di esecuzione ed un certo set di precedenze tecnologiche da rispettare, le quali specificano l'ordine fattibile delle operazioni stesse (3). I principali obiettivi di questo secondo aspetto sono::

Obiettivi di capacità; generalmente i problemi di bilanciamento della linea hanno l'obiettivo di massimizzare la capacità produttiva della stessa, la quale si misura attraverso:

- minimizzazione del numero  $N$  di stazioni;
- minimizzazione del tempo ciclo  $T_c$  ;
- massimizzazione dell'efficienza di bilanciamento;
- minimizzazione della somma dei tempi delle operazioni  $T_p$  ;
- stessi tempi  $T_{si}$  su ogni stazione  $i$ -esima (massimizzazione della Workload Smoothness);
- minimizzazione dei tempi di attesa  $T_{ai}$  ;

Obiettivi di costo; si considerano tutti i costi della linea compresi i costi degli investimenti di lungo periodo e i costi operativi di breve periodo, quindi:

- macchinari e attrezzature;
- manodopera;

- materiali;
- tempi di inattività;
- non completamento;
- set-up (nel multi-model);
- giacenze;

Obiettivi i profitto i quali considerano il rateo produttivo e i prezzi come variabili decisionali;

Obiettivi sociali ed organizzativi (vincoli più che obiettivi):

- job enrichment: arricchimento da parte del lavoratore di nuovi compiti che prevedono un livello di responsabilità superiore a quelli svolti precedentemente;
- job enlargement: aumento delle mansioni affidate ai lavoratori senza però prevedere come nel job enrichment la possibilità di assumere funzioni e poteri diversi da quelli tradizionali.

Il problema decisionale di ottimizzare il bilanciamento del carico di lavoro fra le stazioni rispettando i vincoli di precedenza e il tempo ciclo prende il nome di **Assembly line balancing problem (ALBP)**. Tipicamente ALBP entra in gioco nel momento in cui una impresa introduce una nuova linea di montaggio oppure ne riprogetta una già esistente per la realizzazione di elevati volumi di produzione.

La classificazione più ampia presente in letteratura sui problemi di bilanciamento di una linea di montaggio è la seguente:

- **SALBP** (Simple assembly line balancing problem): classico problema per linee single-model con pre-determinate caratteristiche;
- **GALBP** (General assembly line balancing problem): comprende tutti i problemi che non sono SALB come per esempio il bilanciamento di linee mixed-model, a U, a due lati, parallele ecc.;
- **DLBP** (Design line balancing problem); problema che riguarda la progettazione del bilanciamento di una linea;

- **RALB** (Robotic assembly line balancing): problema del bilanciamento di una linea di montaggio robotizzata.

La maggior parte della ricerca di questa tesi si dedica alla modellazione e risoluzione dei problemi SALBP; si descrivono quindi di seguito le sue caratteristiche principali.

### 2.5.1 SALBP (Simple assembly line balancing problem)

Come già accennato SALBP è il problema che si dedica alla risoluzione del bilanciamento di una classica linea single-model; le caratteristiche del problema sono le seguenti (2):

- produzione di massa di un prodotto omogeneo attraverso l'esecuzione di  $n_e$  operazioni di un certo processo produttivo;
- sistemi di movimentazione automatici con tempo ciclo  $T_c$  fisso;
- tempi delle operazioni deterministici e interi;
- nessuna restrizione sull'assegnazione delle operazioni oltre alle precedenze tecnologiche;
- layout seriale della linea e stazioni unilaterali;
- intervallo di lancio fisso e pari al tempo ciclo.

A seconda della funzione obiettivo SALBP può assumere differenti versioni (riassunte in Tabella 2.2):

- **SALBP-F** (problema di fattibilità): problema decisionale che stabilisce l'esistenza o meno di una linea di montaggio fattibile per una data combinazione del numero  $N$  di stazioni di lavoro e del tempo ciclo  $T_c$ .
- **SALBP-1**: la più famosa versione di SALBP che riguarda la minimizzazione del numero  $N$  di stazioni per un dato tempo ciclo  $T_c$ ;
- **SALBP-2**: versione di SALBP che riguarda la minimizzazione del tempo ciclo  $T_c$  per un dato numero di stazioni  $N$ ;
- **SALBP-E**: versione più generale del problema che mira alla massimizzazione dell'efficienza di bilanciamento attraverso una certa combinazione del numero di stazioni  $N$  e del tempo ciclo  $T_c$ .

<u>Versioni di SALBP</u>		Tempo ciclo	
		Parametro dato	Parametro da minimizzare
<b>Numero di stazioni di lavoro</b>	Parametro dato	<b>SALBP-F</b>	<b>SALBP-2</b>
	Parametro da minimizzare	<b>SALBP-1</b>	<b>SALBP-E</b>

Tabella2.2: Versioni di SALBP

La descrizione dei problemi di ottimizzazione mostra come ciascuno di essi richieda la soluzione di almeno un'istanza SALBP-F. SALBP-1 e SALBP-2 possono inoltre essere considerati duali in quanto il primo considera il tempo ciclo come un vincolo ed è la funzione obiettivo del secondo e viceversa per l'altro.

Dal momento che il rateo produttivo e il tempo ciclo sono parametri fissi, SALBP-1 si presenta nel momento in cui deve essere installata una nuova linea di montaggio; SALBP-2, al contrario, conduce verso la massimizzazione del rateo produttivo di una linea già esistente. Quest'ultimo problema può perciò sorgere quando ci sono dei cambiamenti nel processo produttivo o quando la struttura della domanda richiede una reingegnerizzazione del sistema produttivo.

## 2.6 Metodi per la risoluzione dei problemi SALBP

A questo punto diventa necessario descrivere quali sono i metodi che possono essere utilizzati per risolvere i problemi di bilanciamento di una linea di montaggio.

Una prima macro distinzione si fa tra: metodi **euristici** e metodi **iterativi**.

I primi (dal greco *euristikein* = scoprire), sono metodi (algoritmi) che ricercano soluzioni ammissibili (non necessariamente ottime) di un problema di ottimizzazione; essi hanno quindi la capacità di individuare una soluzione in tempi relativamente brevi, che perciò può non essere ottima.

I secondi invece sono algoritmi che, attraverso un numero teoricamente infinito di passi, si muovono fra approssimazioni successive che man mano convergono alla soluzione esatta in senso limite del problema.

Tutti i problemi di ottimizzazione sorgono nel momento in cui si devono effettuare delle scelte, e risolverli significa trovare una soluzione ottimale fra un numero finito o infinito di alternative. La maggior parte dei problemi di ottimizzazione è NP-hard perciò, nonostante gli sforzi abbiano prodotto importanti progressi negli ultimi anni, non esiste un metodo universale per risolverli. Di conseguenza è aumentato l'interesse per gli algoritmi di approssimazione (iterativi) che possono trovare soluzioni vicine all'ottimo in un tempo di calcolo ragionevole (4).

Di seguito si descrivono brevemente i metodi più trattati ed utilizzati in letteratura.

## 2.6.1 Metodi euristici

### 2.6.1.1 Largest candidate rule (LCR)

Il procedimento di questo metodo prevede i seguenti passaggi:

- ordinare le operazioni rispetto al tempo di esecuzione, in ordine decrescente;
- assegnare ciascuna operazione elementare ad una stazione di lavoro a partire dalla prima nella lista in modo tale da rispettare il  $\max\{T_{si}\}$  per ciascuna stazione *i-esima* ed i vincoli di precedenza;
- nel momento in cui non è più possibile assegnare le operazioni ad una stazione si passa alla stazione successiva;
- ripetere i passi precedenti fino ad esaurire le operazioni.

Considerando l'esempio del phon del paragrafo 2.4.1, si esegue il metodo LCR. Si ordinano le operazioni elementari in ordine decrescente e si ottiene:

Operazioni	Tempo di esecuzione [min]	Precedenze
S	0,58	R,O
E	0,58	C,D
G	0,42	F
M	0,37	L
B	0,35	A
Q	0,35	P
H	0,33	-

I	0,33	H,G
O	0,235	M,N
L	0,196	I
D	0,15	-
P	0,133	M
R	0,112	Q
N	0,108	-
C	0,105	B
A	0,1	-
F	0,1	E

Tabella 2.3: Operazioni elementari dell'esempio 2.4.1 in ordine decrescente

Seguendo i passi elencati si procede con le assegnazioni delle operazioni nelle stazioni di lavoro e si ottiene il seguente risultato:

Stazione	Operazioni	$T_{max}$	$T_{si}$	$T_{ai}$
1	H,D,N	0,66	0,588	0,072
2	A,B,C	0,66	0,555	0,105
3	E	0,66	0,58	0,08
4	F,G	0,66	0,52	0,14
5	I,L	0,66	0,526	0,134
6	M,O	0,66	0,605	0,055
7	P,Q,R	0,66	0,595	0,065
8	S	0,66	0,58	0,08

Tabella 2.4: Assegnazione delle operazioni (esempio 2.4.1) alle stazioni di lavoro con metodo LCR

### 2.6.1.2 Kilbridge e Wester (K&W)

Il procedimento di questo metodo prevede i seguenti passaggi:

- dividere il grafo delle precedenze in colonne all'interno delle quali sono disposti gruppi di operazioni, in modo tale da ottenere una lista ordinata secondo le precedenze, e, nell'ambito di ciascuna colonna, secondo il tempo di esecuzione;
- assegnare ciascuna operazione elementare ad una stazione di lavoro a partire dalla prima nella lista in modo tale da rispettare il  $\max\{T_{si}\}$  per ciascuna stazione  $i$ -esima ed i vincoli di precedenza;
- nel momento in cui non è più possibile assegnare le operazioni ad una stazione si passa alla stazione successiva;

- ripetere i passi precedenti fino ad esaurire le operazioni.

Anche per questo metodo si mostra come si eseguono i passi facendo riferimento all'esempio del phon del paragrafo 2.4.1. Di seguito (Figura 2.11) si presenta il grafo suddiviso in colonne.

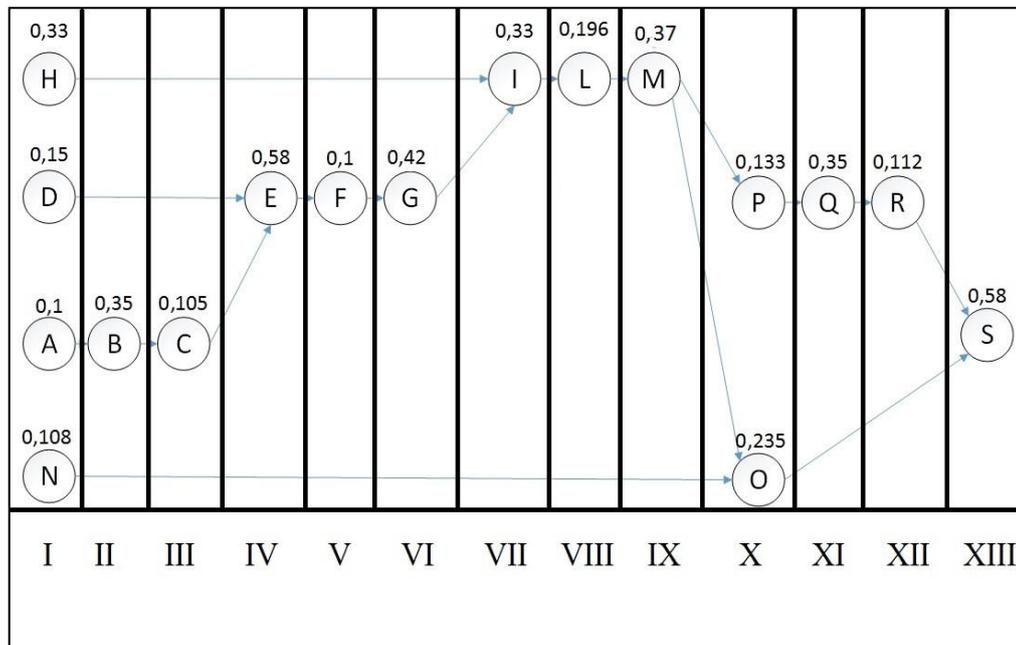


Figura 2.11: Suddivisione del grafo in colonne (esempio 2.4.1)

Da questa rappresentazione si ottiene la seguente lista:

Operazioni	Colonna	Tempo di esecuzione [min]	Precedenze
H	1	0,33	-
D	1	0,15	-
A	1	0,1	-
N	1	0,108	-
B	2	0,35	A
C	3	0,105	B
E	4	0,58	C,D
G	6	0,42	F
I	7	0,33	G,H
L	8	0,196	I
M	9	0,37	L
O	10	0,235	M,N
P	10	0,133	M
Q	11	0,35	P
R	12	0,112	Q
S	13	0,58	O,R

Tabella 2.5: Suddivisione delle operazioni nelle colonne

In questo caso, questa lista porta alla medesima soluzione ottenuta con il metodo LCR.

Stazione	Operazioni	$T_{max}$	$T_{si}$	$T_{ai}$
1	H,D,A	0,66	0,58	0,08
2	N,B,C	0,66	0,563	0,097
3	E	0,66	0,58	0,08
4	F,G	0,66	0,52	0,14
5	I,L	0,66	0,526	0,134
6	M,O	0,66	0,605	0,055
7	P,Q,R	0,66	0,595	0,065
8	S	0,66	0,58	0,08

Tabella 2.6: Assegnazione delle operazioni (esempio 2.4.1) alle stazioni di lavoro con metodo K&amp;W

### 2.6.1.3 Ranked positional weight (RPW)

Il procedimento di questo metodo prevede i seguenti passaggi:

- calcolare, per ogni operazione elementare, l'indice  $RPW$  dato dalla somma del tempo di esecuzione dell'operazione stessa più quelli delle operazioni che la seguono nel grafo delle precedenze; si ottiene una lista ordinata per  $RPW$  decrescenti;
- assegnare ciascuna operazione elementare ad una stazione di lavoro a partire dalla prima nella lista in modo tale da rispettare il  $\max\{T_{si}\}$  per ciascuna stazione  $i$ -esima ed i vincoli di precedenza;
- nel momento in cui non è più possibile assegnare le operazioni ad una stazione si passa alla stazione successiva;
- ripetere i passi precedenti fino ad esaurire le operazioni.

Si applica il metodo all'esempio del phon, paragrafo 2.4.1. dal grafo delle precedenze si ottiene la seguente lista:

Operazioni	Tempo di esecuzione [min]	Precedenze	RPW
A	35	-	3,961
B	0,58	A	3,861

D	0,15	-	3,556
C	0,105	A-B	3,511
E	0,58	D-C	3,406
F	0,1	E	2,826
G	0,42	F	2,726
H	0,33	-	2,636
I	0,33	G-H	2,306
L	0,196	I	1,976
M	0,37	L	1,78
P	0,133	M-N	1,175
Q	0,35	P	1,042
N	0,108	-	0,923
O	0,235	M-N	0,815
R	0,112	Q	0,692
S	0,58	O-R	0,58

Tabella 2.7: Calcolo dell'indice RPW dell'esempio del phon (2.4.1)

Stazione	Operazioni	$T_{max}$	$T_{si}$	$T_{ai}$
1	A,B,D	0,66	0,6	0,06
2	C,H,N	0,66	0,543	0,117
3	E	0,66	0,58	0,08
4	F,G	0,66	0,52	0,14
5	I,L	0,66	0,526	0,134
6	M,P	0,66	0,553	0,107
7	Q,O	0,66	0,585	0,075
8	R	0,66	0,112	0,6488
9	S	0,66	0,58	0,08

Tabella 2.6: Assegnazione delle operazioni (esempio 2.4.1) alle stazioni di lavoro con metodo RPW

Questo metodo genera una soluzione con un numero di stazioni di lavoro superiore a quello dei due metodi precedenti.

Uno dei limiti più grossi dei metodi euristici è quello che essi permettono di generare una sola soluzione, che non è detto che sia la migliore (5). I metodi iterativi che si descrivono di seguito, partono invece da una o più soluzioni possibili e giungono alla soluzione ottima.

## 2.6.2 Metodi iterativi

### 2.6.2.1 Tabu Search

La Tabu Search è un'evoluzione del modello “metodo di discesa”, utilizzato per trovare il minimo di una funzione all'interno di uno spazio di soluzioni. Il “metodo di discesa” consiste nel partire da una soluzione iniziale e da questa eseguire una serie di mosse che portano ad una nuova soluzione all'interno dell'insieme di adiacenza della soluzione corrente, la cui funzione obiettivo assume un valore minimo inferiore a quello attuale. Il difetto di tal metodo consiste nel fatto che, se nell'insieme di adiacenza non esistono soluzioni migliori a quella corrente, la ricerca si ferma. Questo significa che la soluzione ottima risulta essere un minimo locale dello spazio delle soluzioni, spesso lontana dalla soluzione ottima globale.

La tecnica della Tabu Search nasce con l'obiettivo di continuare la ricerca oltre ai minimi locali. Il concetto di base consiste nel rendere proibite (‘tabu’) le ultime mosse eseguite nel cammino di ricerca, in modo tale da evitare la trappola dei minimi locali. Caratteristica fondamentale è perciò l'uso della memoria: durante il procedimento viene tenuta traccia non solo delle informazioni locali, ma anche di informazioni relative all'itinerario percorso. Tali informazioni vengono impiegate per procedere dalla soluzione corrente a quella successiva che viene scelta all'interno dell'insieme di adiacenza.

La ricerca può terminare in base a differenti criteri di stop, ovvero quando:

- si raggiunge il numero massimo di iterazioni;
- si raggiunge il tempo massimo di calcolo;
- il valore della funzione obiettivo trovato è uguale al limite inferiore.

### 2.6.2.2 Simulated Annealing

La Simulated Annealing è una tecnica di ricerca tipicamente utilizzata nella risoluzione dei problemi di ottimizzazione; in particolare essa cerca di evitare di rimanere intrappolata in soluzioni ottime locali nella ricerca di quelle globali.

Il nome della tecnica deriva dal concetto di “ricottura” nella scienza dei metalli, usato per descrivere il processo di eliminazione di difetti reticolari dai cristallini tramite una procedura di riscaldamento seguita da un lento raffreddamento; lo scopo della ricottura delle particelle del solido è quello di far sì che questo possa raggiungere alcuni attributi desiderati, come per esempio resistenza o durezza superficiale.

Analogamente al suo uso nella ricottura fisica dei solidi, la soluzione del problema di ottimizzazione subisce una serie di modifiche durante la ricerca di una soluzione migliore (secondo alcune funzioni obiettivo). Dal momento che la tecnica inizia, si genera una prima soluzione che si considera come prima soluzione corrente; si definisce un parametro di controllo  $T$  analogo alla temperatura di ricottura. Tale “temperatura” viene sistematicamente diminuita, o “raffreddata”, a seconda di una certa velocità di raffreddamento  $CR$ . Quando la temperatura scende, si individuano soluzioni confinanti alla soluzione corrente e, se il valore della funzione obiettivo è superiore a quello della funzione attuale, la soluzione confinante diventa la nuova soluzione corrente. Se invece la soluzione confinante fornisce un valore inferiore rispetto a quella corrente delle funzione obiettivo, essa può ancora diventare la soluzione attuale, ma solo se soddisfa un determinato criterio di accettazione. Infatti, una delle caratteristiche fondamentali della Simulated Annealing è proprio quella di accettare soluzioni inferiori come correnti per evitare di rimanere intrappolati in ottimi locali. Attraverso l'accettazione occasionale di soluzioni inferiori che soddisfano i criteri di accettazione, la ricerca si muove sul continuum di soluzioni realizzabili, nel tentativo di raggiungere l'ottimo globale. Il processo si ripete fino a che non si raggiungono determinati criteri di stop.

Entrambi i metodi iterativi citati sopra vengono applicati in ambito di risoluzione dei problemi di bilanciamento di una linea di assemblaggio e i loro principali obiettivi sono:

- la minimizzazione del numero di stazioni di lavoro;
- la saturazione delle stazioni di lavoro, ovvero la minimizzazione dei tempi di attesa delle stazioni.

Il limite dei metodi iterativi appena descritti consiste nel fatto che essi non permettono l'ottimizzazione dei problemi di montaggio sulla base di più obiettivi, oltre a quelli appena citati. Questo vincolo si può invece abbattere utilizzando l'algoritmo genetico che mette a

disposizione una funzione di valutazione che può essere multi obiettivo, e che quindi è in grado di mettere in gioco parametri nuovi e altrettanto importanti nel bilanciamento di una linea.

### **2.6.2.3 Algoritmo genetico**

L'algoritmo genetico è un algoritmo che si ispira al principio di evoluzione naturale ed evoluzione biologica introdotto da Darwin; il termine "genetico" deriva dal fatto che tale algoritmo attua dei meccanismi concettualmente simili a quelli dei processi biochimici scoperti da questa scienza.

In pratica esso permette di valutare delle soluzioni di partenza dopodiché, ricombinandole ed introducendo degli elementi di disordine, è in grado di crearne nuove nel tentativo di convergere a soluzioni ottime (6). Tipicamente tali algoritmi vengono utilizzati per risolvere problemi di ottimizzazione.

L'algoritmo genetico rappresenta lo strumento che si utilizza in questo lavoro; se ne parlerà quindi più approfonditamente nei prossimi capitoli al fine di ottimizzare le soluzioni di assemblaggio.

## Capitolo 3 Algoritmo genetico per la risoluzione di SALBP

### Terminologia generale di un algoritmo genetico

Cromosoma: una delle possibili soluzioni relative ad un problema considerato;

Popolazione: insieme di soluzioni relative al problema considerato;

Gene: parte di un cromosoma;

Funzione di Fitness: grado di valutazione associato ad una soluzione. La valutazione avviene in base ad una funzione appositamente progettata detta Funzione di Fitness o funzione obiettivo;

Crossover: operatore genetico che prevede la generazione di una nuova soluzione che si ottiene con la ricombinazione di soluzioni già esistenti;

Mutazione: operatore genetico che prevede l'alterazione casuale di una soluzione.

### 3.1 Cosa è un algoritmo genetico e come viene utilizzato per risolvere i problemi di montaggio

Come già accennato, l'algoritmo genetico è un algoritmo di approssimazione che, ispirandosi alla teoria dell'evoluzione di Darwin (7) (ovvero il processo che controlla l'evoluzione biologica degli organismi noto come selezione naturale e riproduzione), è in grado di individuare soluzioni vicine all'ottimo in un tempo di calcolo ragionevole (4). In particolare:

- la selezione naturale determina quali organismi hanno l'opportunità di riprodursi e di sopravvivere all'interno della popolazione;
- la riproduzione coinvolge geni di due diversi individui (chiamati genitori) che si combinano per formare figli i quali ereditano le loro caratteristiche.

L'algoritmo necessita di una semplice funzione obiettivo per valutare la qualità delle soluzioni candidate; inoltre le procedure di valutazione della qualità di una soluzione possono essere facilmente cambiate o modificate, fornendo una flessibilità tale da prendere in considerazione tutti gli elementi ed i fattori che riguardano l'assemblaggio e il bilanciamento di una linea (8).

Più nel dettaglio, l'algoritmo genera iterativamente popolazioni di soluzioni fino a che non raggiunge la soluzione che rappresenta il miglior risultato. Attraverso la generazione di ogni successiva popolazione l'algoritmo ottiene miglioramenti nella qualità di ogni singola soluzione; così facendo è in grado di muoversi verso risultati positivi senza aver bisogno di esaminare ogni possibile soluzione del problema.

L'algoritmo genetico sta ricevendo molta attenzione da parte dei ricercatori nella risoluzione dei problemi di montaggio in quanto eccelle per la sua capacità di trovare soluzioni ottime rispetto alle tradizionali tecniche di ottimizzazione (3).

In circa 29 articoli (dal 1992 al 2006) che trattano gli algoritmi genetici rispetto a differenti modi di codificare i cromosomi e a come gli operatori genetici sono utilizzati per la riproduzione, si nota che 14 si focalizzano sulla risoluzione di problemi di tipo SALPB; i restanti invece trattano gli altri problemi del bilanciamento come per esempio quelli riguardanti le linee mixed-model (4).

Uno studio (4) propone anche un'analisi approfondita di quei 29 algoritmi rispetto a nove criteri fondamentali per il raggiungimento di una buona performance dell'algoritmo stesso. I nove criteri sono: il metodo di algoritmo proposto, la generazione della popolazione iniziale, la rappresentazione genetica degli individui, la valutazione della funzione obiettivo, gli operatori genetici scelti per alterare le soluzioni, la selezione degli individui per la generazione, il tipo di sopravvivenza per la nuova generazione e i criteri di stop.

Questi criteri riassumono in pratica gli step che devono essere percorsi per completare l'algoritmo.

I principali passi dell'algoritmo si possono schematizzare come segue:

1. rappresentazione genetica e generazione delle popolazione iniziale di soluzioni (cromosomi);
2. valutazione della funzione di fitness per le soluzioni che appartengono all'attuale popolazione;

3. processo di riproduzione attraverso la selezione e gli operatori genetici di crossover e mutazione;
4. riparazione dei cromosomi non fattibili ed inserimento dei nuovi cromosomi nella popolazione;
5. iterazione: se il predeterminato criterio di stop è stato raggiunto allora selezione e decodifica del miglior cromosoma, altrimenti riesecuzione della procedura a partire da 2 utilizzando la nuova popolazione creata in 4.

L'iterazione di tali passi permette l'evoluzione verso una soluzione ottimizzata del problema considerato.

Nella letteratura esistono differenti possibilità di eseguire ciascun passo dell'algoritmo, ognuno dei quali può essere caratterizzato da pro e contro; sta all'utilizzatore capire quali sono le tipologie per ciascun passo che meglio si adattano al caso in esame.

## **3.2 Analisi dello stato dell'arte**

Si focalizza l'attenzione, per lo studio dello stato dell'arte, nell'analisi di tutti i casi che riguardano i problemi SALBP, con particolare interesse a quelli del primo tipo che sono argomento di studio di questo elaborato; si tiene poi conto di quelli che si riferiscono alla risoluzione di problemi diversi da SALBP, i cui obiettivi però sono considerati significanti per la presente ricerca.

Nei prossimi paragrafi si procede con una descrizione dettagliata di tutti i passi dell'algoritmo.

### **3.2.1 Rappresentazione genetica e generazione della popolazione iniziale di soluzioni**

La rappresentazione genetica ha il compito di definire un individuo (soluzione) in modo univoco ed è il primo passo applicato dall'algoritmo che ha lo scopo di convertire la soluzione in una stringa chiamata cromosoma. Nella letteratura applicata al bilanciamento di una linea si parla di diversi tipi di rappresentazioni per un cromosoma:

- “*Task based representation*” (Figura 3.1), nella quale i cromosomi sono definiti come una sequenza di operazioni fattibili, ovvero che rispettano le precedenze tecnologiche;

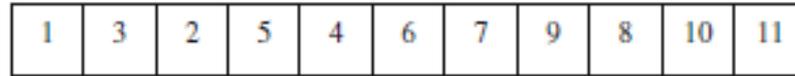


Figura 3.1: “*Task based representation*”

in pratica si elencano le operazioni elementari nell’ordine in cui vengono assegnate alle stazioni (9);

- “*Embryonic representation*” (Figura 3.2) (10), è una particolare versione della precedente. L’unica differenza sta nel fatto che in questo caso vengono considerati i sottoinsiemi di soluzioni piuttosto che le singole soluzioni quindi la lunghezza del cromosoma può variare da generazione a generazione;

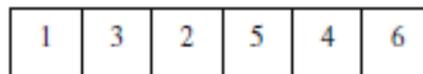


Figura 3.2: “*Embryonic representation*”

- “*Workstation based representation*” (Figura 3.3), nella quale il cromosoma è definito come un vettore che contiene le etichette delle stazioni di lavoro alle quali sono assegnate le operazioni. La lunghezza del cromosoma è perciò definita dal numero di operazioni;

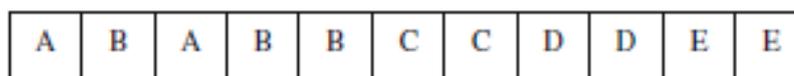


Figura 3.3: “*Workstation based representation*”

- “*Grouping based representation*” (Figura 3.4) (11), nella quale la codifica si divide in due parti, la prima è “*workstation based*” e la seconda “*group based*”. La “*group part*” della codifica ha il semplice compito di mettere in lista tutte le stazioni di lavoro della soluzione corrente; viene posta dopo una colonna che la divide dalla prima codifica. La lunghezza dei cromosomi varia da soluzione a soluzione;

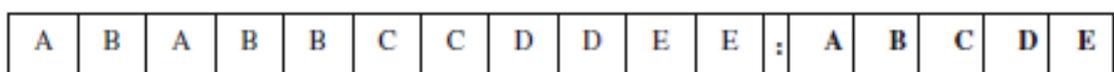


Figura 3.4: “*Grouping based representation*”

La rappresentazione genetica costituisce uno step cruciale nel quale si descrivono le caratteristiche del problema, ha perciò una forte influenza sui passi successivi dell'algoritmo.

Per quanto riguarda la generazione della popolazione iniziale, nel caso di SALBP-1, una soluzione rappresenta la distribuzione delle operazioni fra le  $N$  stazioni. I cromosomi possono rappresentare soluzioni fattibili o ammissibili e tipicamente sono circa dieci volte superiori al numero di operazioni richieste per un particolare montaggio.

Una soluzione si considera fattibile se la somma dei tempi delle operazioni su una stazione è inferiore al tempo ciclo e se vengono rispettate le precedenze; una soluzione è ammissibile se vale solo il primo vincolo dei due appena descritti.

La popolazione iniziale degli individui si genera in modo casuale e si evolve attraverso le iterazioni dell'algoritmo chiamate generazioni.

### **3.2.2 Valutazione della funzione di fitness per le soluzioni che appartengono all'attuale popolazione**

La fitness di una soluzione rappresenta la funzione obiettivo che deve essere ottimizzata. In poche parole essa identifica la bontà della soluzione e la probabilità che un individuo venga scelto come genitore è proporzionale alla sua fitness quindi è più probabile che venga scelto un individuo con fitness più alta.

Dal momento che le soluzioni generate provengono dalle caratteristiche delle soluzioni padri, ci aspettiamo che la nuova popolazione abbia una qualità media superiore a quella di partenza.

Si dedicherà l'intero paragrafo 3.2.6 alla discussione di come in letteratura sono utilizzate le fitness function, con particolare riferimento agli obiettivi che esse cercano di ottimizzare.

### 3.2.3 Processo di riproduzione attraverso la selezione e gli operatori genetici di crossover e mutazione

Vediamo come si genera la nuova popolazione a partire da quella attuale. Si considera il numero di individui (soluzioni) che appartengono alla popolazione attuale; a questo punto il processo utilizzato per generare la nuova generazione fa in modo che quest'ultima abbia la stessa dimensione di quella di partenza. Così facendo, la dimensione della popolazione ad ogni iterazione dell'algoritmo rimane invariata.

Si applica quindi la selezione, dopodiché il crossover o la mutazione, a seconda della probabilità di accadimento che si è stabilita per ognuno dei due. Nella maggior parte dei casi la probabilità stabilita per l'operatore genetico crossover è molto superiore a quella stabilita per la mutazione (95-98 % per il primo operatore e meno dell'1 % per il secondo).

Di seguito si mostrano nel dettaglio le tre operazioni del processo di riproduzione e per ognuna di esse si descrivono le tipologie più utilizzate in letteratura:

#### Selezione

La selezione è una tecnica che dirige la ricerca verso regioni promettenti all'interno dello spazio delle soluzioni. Si presentano tre esempi delle diverse tecniche di selezione:

- “*Roulette wheel selection*”, tecnica con la quale si calcola per ogni cromosoma la probabilità che esso ha di essere selezionato; tale probabilità è proporzionale al valore della sua fitness

$$p_i = f_i / \sum f_k \quad k = 1, \dots, n$$

Maggiore è il valore della fitness che un cromosoma possiede, maggiori sono le chances che ha di essere selezionato. Due individui perciò sono scelti in modo casuale, tenendo conto però sulle loro probabilità. Il processo di selezione si basa sul “girare” la ruota (wheel) un numero di volte pari alla dimensione della popolazione (3);

- “*Selezione per categoria*”, simile alla precedente ma in questo caso la valutazione si effettua in modo proporzionale alla somma del valore della funzione di fitness per ogni coppia di possibili soluzioni. Con questa tecnica si può presentare il

problema della lentezza della convergenza nel caso in cui le differenze fra le coppie selezionate siano troppo piccole;

- “*Selezione a torneo*” è una tecnica nella quale si raggruppano le soluzioni e si valutano con il seguente algoritmo:
  - scegliere in maniera casuale  $n$  individui appartenenti alla popolazione;
  - scegliere l'individuo migliore e impostare la sua probabilità di scelta a  $p$ ;
  - scegliere il secondo individuo migliore e impostare la probabilità di scelta a  $p * (1 - p)$ ;
  - scegliere il terzo individuo migliore e impostare la sua probabilità di scelta a  $p * (1 - p)^2$ ;

Poi proseguire fino ad esaurimento delle soluzioni scelte.

### Crossover

È il principale operatore genetico che lavora su due cromosomi (genitori) al fine di produrre due nuovi cromosomi (figli) scambiando i geni. I principali operatori di crossover sono:

- “*One point crossover*” (crossover ad un punto, Figura 3.5), nel quale si seleziona un punto casuale o predefinito che appartiene ai due cromosomi padre, così da dividerli in due parti (testa e coda). La prima nuova soluzione sarà data dalla combinazione della testa delle prima soluzione e dalla coda della seconda e la seconda nuova soluzione viceversa.

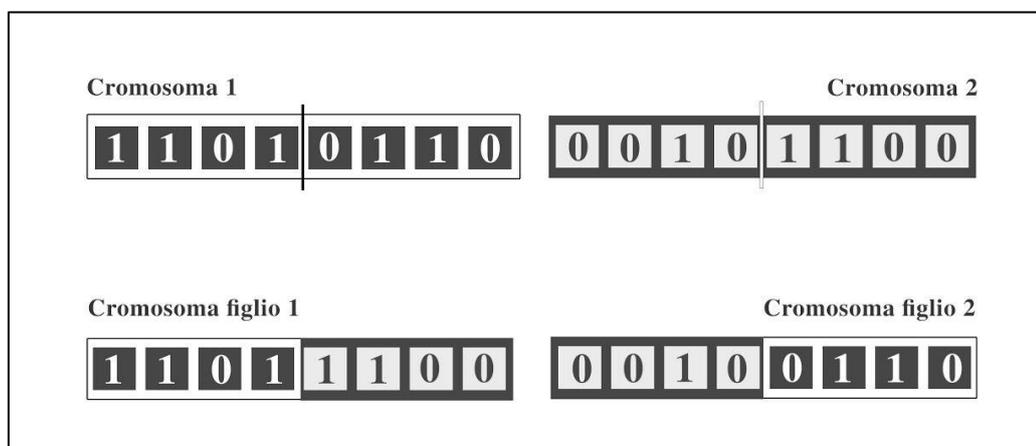


Figura 3.5: “*One point crossover*”

- “*Two point crossover*” (crossover a due punti), nel quale si seleziona due punti casuali o predefiniti che appartengono ai due cromosomi padre, così da dividerli in tre parti (testa, parte centrale e coda). La prima nuova soluzione sarà data dalla testa e dalla coda della prima soluzione e dalla parte centrale della seconda e la seconda nuova soluzione viceversa.  
All’interno di questa tipologia di crossover si distinguono per esempio il “*Partially Mapped Crossover*” oppure l’”*Order Crossover*” i quali, nonostante appartengano allo stesso operatore genetico, provocano effetti molto dissimili (12);
- “*Uniform order based crossover*” (Figura 4.9 del capitolo 4), nel quale si selezionano due punti casuali che appartengono ai due cromosomi padri al fine di ottenere una testa, una coda e una parte centrale della prima e della seconda soluzione. La prima nuova soluzione sarà data dalla testa e dalla coda del primo cromosoma padre e dalla parte centrale formata dai geni mancanti così come appaiono nella seconda soluzione padre. La seconda nuova soluzione sarà data dalla testa e dalla coda del secondo cromosoma padre e dalla parte centrale dei geni mancanti così come appaiono nella prima soluzione padre.

Risultati di esperimenti dimostrano che la migliore nuova soluzione che può venire fuori da due genitori attraverso l’”*Uniform order based crossover*” è almeno buona quanto quella che può venire fuori attraverso il “*Two point crossover*” che a sua volta è almeno buona quanto quella che può venire fuori attraverso il “*One point crossover*” (12).

### Mutazione

La mutazione è un operatore genetico (Figura 3.6) utilizzato per mantenere la diversità genetica fra una generazione e quella successiva e quindi per non cadere in ottimi locali. La mutazione opera su un singolo cromosoma andando a scambiare i geni del cromosoma stesso.



Figura 3.6: Operatore genetico mutazione

I principali operatori di mutazione sono:

- “*Modified Mutation*”, usata principalmente per le rappresentazioni “*Grouping based representation*” ed applicata solamente alla parte del gruppo del cromosoma mentre la prima parte della codifica (“*workstation based*”) rimane la stessa;
- “*Heuristic structural Mutation*”, la quale seleziona casualmente alcune operazioni da ogni cromosoma, in modo proporzionale alla probabilità di mutazione, e poi le riassegna a quei cromosomi;
- “*Scramble Sublist Mutation*”, la quale seleziona una sottolista da un cromosoma prendendo la parte centrale determinata casualmente da due punti. Gli elementi della sottolista vengono poi permutati; la nuova soluzione quindi ha la parte centrale circondata dai dati originali costituiti dagli estremi del cromosoma padre.

### 3.2.4 Riparazione dei cromosomi non fattibili ed inserimento dei nuovi cromosomi nella popolazione

Un aspetto importante nella progettazione di un algoritmo genetico è quello di decidere se gli individui generati dagli operatori genetici che non risultano fattibili possono essere ammessi o no alla popolazione. Attraverso la letteratura si possono evidenziare tre strategie principali che fanno fronte a questo problema:

- la prima forza gli individui alla fattibilità creando già dalla prima iterazione in modo casuale solo individui fattibili;

- la seconda ripara gli individui infattibili nei punti in cui i vincoli di precedenza tecnologica non vengono rispettati;
- la terza strategia inserisce nel calcolo della funzione di fitness degli individui non fattibili una penalità, in modo che questi non vengano selezionati per l'evoluzione e scartati attraverso le iterazioni dell'algoritmo.

### 3.2.5 Selezione e procedura finale

A valle del processo di riproduzione si ottiene una popolazione (genitori più figli) che ha dimensione pari al doppio di quella di partenza. Fra i cromosomi si seleziona perciò quelli con la fitness migliore i quali comporranno la nuova generazione di dimensioni pari a quella di partenza.

Per evitare di perdere i migliori cromosomi a causa della selezione, si può utilizzare un metodo chiamato elitismo che evita tale possibilità.

L'”elitismo” permette di selezionare gli individui che presentano una fitness maggiore di una certa soglia, impostata da configurazione. E' un metodo che prima copia il miglior cromosoma (o i pochi migliori) nella nuova popolazione, e poi effettua il resto della selezione nella maniera classica; è quindi in grado di far crescere rapidamente la performance degli algoritmi genetici perché evita la perdita di buone soluzioni (13).

Si reitera la procedura dal passo 2 al passo 5 per ottenere una popolazione costituita dai cromosomi con la più alta fitness, fino alla scelta del cromosoma migliore che rappresenterà la sequenza di montaggio che meglio bilancia la linea.

Il processo evolutivo si ripete fino a che non si raggiungono determinati criteri di stop. Solitamente si tiene conto di uno dei seguenti criteri oppure di una loro combinazione (7):

- è richiesto all'utente di definire il numero di iterazioni che l'algoritmo deve eseguire e la ricerca si arresta quando viene raggiunto tale numero;
- viene definito a priori un tempo di processazione massimo dell'algoritmo;
- il sistema si arresta quando si raggiunge la convergenza delle soluzioni ovvero quando, dopo un certo numero di iterazioni, non si ottengono più soluzioni migliori e la funzione di fitness si appiattisce.

### 3.2.6 Fitness Function

Gran parte dell'analisi dello stato dell'arte dell'algoritmo genetico si dedica allo studio delle fitness function.

Come già accennato, la fitness function rappresenta la bontà della soluzione e la probabilità che un individuo venga scelto come genitore è proporzionale alla sua fitness; si dimostra perciò ovvia la sua importanza nella risoluzione dell'algoritmo.

In letteratura si sono individuate svariate funzioni, le quali differiscono principalmente per la tipologia di problema che vanno a trattare e per l'obiettivo (singolo o multiplo).

In questo testo ci siamo soffermati sullo studio delle fitness function, ed in generale degli algoritmi genetici, che si propongono per la risoluzione dei problemi ALBP ed in particolar modo SALBP.

Di seguito si riporta la lista, in ordine cronologico, delle fitness function individuate (per ognuna si specifica il nome dell'articolo, l'anno di pubblicazione e l'obiettivo); per comodità si utilizza la simbologia trovata per ogni funzione (con relativa spiegazione dei termini che la compongono).

1. A genetic algorithm for grouping (1991) (11)

$$f(S) = \frac{\sum_{k=1}^{UM} (t(S_k)/c)^2}{UM}$$

dove  $t(S_k)$  rappresenta il tempo necessario all'operatore della stazione  $k$ -esima di completare l'operazione di montaggio,  $c$  il tempo ciclo e  $UM$  il limite superiore del numero di stazioni;

2. A genetic algorithm for bin packing and line balancing (1992) (12)

Questo utilizza la stessa fitness function di 1;

3. Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria (1994) (15)

$$f_1(S) = \sum_{i=1}^n (c - t(W S_i))^2 / n$$

$$f_2(S) = \sum_{i=1}^n (c - t(W S_i)) / n$$

$$f_3(S) = 2\sqrt{f_1(S)} + f_2(S)$$

dove  $c$  rappresenta il tempo ciclo e  $W S_i$  il carico di lavoro della stazione  $i$ -esima e  $n$  il numero di stazioni di lavoro;

4. Solving fuzzy Assembly-line Balancing Problem with Genetic Algorithms (1995) (16)

$$eval(v_k) = \sum_{i=1}^m (c_k - t_{ik})$$

dove  $c_k$  rappresenta il tempo ciclo e  $t_{ik}$  il tempo necessario a completare tutte le operazioni sulla stazione  $W_i$  per ogni cromosoma  $v_k$  e  $m$  il numero di stazioni;

5. A hybrid grouping algorithm for bin packing (1996) (17)

Questo utilizza la stessa fitness function di 1;

6. Genetic Algorithms for assembly line balancing with various objectives (1996) (18)

In questo articolo le fitness sono rappresentate dalle seguenti funzioni obiettivo da ottimizzare:

- massimizzazione della distribuzione del carico di lavoro fra le stazioni (Workload Smoothness):

$$\text{Minimize } f = \frac{1}{N} \sum_{j=1}^N \left( T_j - \frac{W}{N} \right)^2$$

dove  $N$  rappresenta il numero di stazioni,  $T_j$  il carico della stazione  $j$ -esima e  $W$  il tempo totale di processazione delle operazioni

- massimizzazione della capacità di assegnare le operazioni in modo tale da far sì che quelle che sono fra loro collegate siano inserite all'interno della stessa stazione di lavoro (Workload Relatedness):

$$\text{Minimize } f = n - \frac{n}{\sum_{j=1}^n SN_j}$$

dove  $n$  rappresenta il numero di stazioni come variabile decisionale e  $SN_j$  il numero di archi connessi in  $G_j$  (rete che rappresenta le relazioni di precedenza delle operazioni assegnate alla stazione  $j$ -esima).

- utilizzo congiunto di due funzioni, ovvero la massimizzazione della Workload Smoothness e la massimizzazione della Workload Relatedness:

$$\text{Minimize } f(1) = \frac{1}{N} \sum_{j=1}^N \left( T_j - \frac{W}{N} \right)^2 \text{ e } \text{Minimize } f(2) = \sum_{j=1}^n SN_j;$$

#### 7. Fuzzy Assembly line balancing using genetic algorithms (1996) (19)

Questo utilizza la stessa fitness function di 4;

#### 8. Multi-objective genetic algorithm and its applications to flowshop scheduling (1996) (20)

$$F(x) = w_1 f_1(x) + \dots + w_n f_n(x)$$

dove:

- $F(x)$  è la funzione di fitness combinata
- $f_i(x)$  è la  $i$ -esima funzione di fitness (da massimizzare)
- $w_i$  è il peso costante per  $f_i(x)$
- $n$  è il numero di funzioni obiettivo

Se si utilizzano pesi costanti per calcolare la funzione, anche la direzione della ricerca nell'algoritmo genetico è costante. Quindi, l'idea proposta dal ricercatore è

quella di utilizzare pesi  $w_i$  che si generano casualmente utilizzando la seguente relazione:

$$w_i = \frac{RN_i}{\sum_{j=1}^n RN_j}$$

dove  $RN_i$  e  $RN_j$  sono interi casuali non negativi. Gli  $n$  numeri casuali reali generati per i pesi  $w_i$  sono usati per calcolare la somma pesata  $F(x)$ , dal momento che ogni paio di soluzioni viene selezionato. La somma pesata  $F(x)$  è usata per calcolare la probabilità di selezione di ogni soluzione, che in questo modo è variabile per ogni singola generazione;

9. A grouping genetic algorithm for line balancing with resource dependent task times (1997) (21)

Questo utilizza la stessa fitness function di 1;

10. A Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling (1998) (22)

Questo utilizza la stessa fitness function di 8;

11. A Heuristic-based genetic algorithm for workload smoothing in assembly lines (1998) (23)

$$eval = \frac{1}{n} \sum_{j=1}^n \sqrt{|T_j - \bar{T}|}$$

dove  $T_j = \sum_{i \in t_j} t_i$  ( $t_i$  è il tempo di processazione dell'operazione  $i$ -esima) è il carico della stazione  $j$ -esima, dove  $\bar{T} = \frac{1}{n} \sum_{i \in I} t_i$  è il carico medio e  $n$  il numero di stazioni;

12. Assembly line balancing using genetic algorithms (2000) (24)

$$f(S) = 2 \sqrt{\frac{\sum_{k=1}^{UM} (tS_{max} - t(S_k))^2}{UM}} + \frac{\sum_{k=1}^{UM} (tS_{max} - t(S_k))}{UM}$$

dove  $tS_{max}$  è il massimo tempo della stazione ( $tS_{max} \leq c$ ),  $t(S_k)$  il tempo necessario all'operatore della stazione  $k$ -esima di completare l'operazione di montaggio e  $UM$  il limite superiore del numero di stazioni; il primo termine dell'equazione ha lo scopo di ridurre lo squilibrio, e il secondo termine quello di minimizzare il numero di stazioni;

13. A multi-Objective Genetic Algorithm for Solving Assembly line Balancing Problem (2000) (25)

Questo utilizza la stessa fitness function 8;

14. Local Search Heuristics for the Assembly Line Balancing Problem with Incompatibilities Between Tasks (2000) (26)

$$f(S) = \frac{\sqrt{\sum_{k=1}^{UM} (c - t(S_k))^2}}{c * \sqrt{UM}} + LM1 - m_s$$

dove  $LM1 := \lceil t_{sum} / c \rceil = \lceil \sum_{j=1}^n p_j \rceil$  e  $m_s$  è il numero delle stazioni della soluzione corrente. Il primo termine considera il livello di squilibrio;

15. A hybrid genetic algorithm approach on multi-objective of assembly planning problem (2002) (25)

$$FV = \sum_{i=1}^5 w_i f_i$$

dove:

- $f_1$  rappresenta la minimizzazione del tempo ciclo
- $f_2$  rappresenta la massimizzazione della distribuzione del carico di lavoro fra le stazioni (Workload Smoothness)
- $f_3$  rappresenta la frequenza di cambio dei tools

- $f_4$  rappresenta il numero di tools
- $f_5$  rappresenta la pena totale delle operazioni di assemblaggio

La pena rappresenta il costo che si incorre nel momento in cui si ha una sequenza di assemblaggio non idonea o infattibile;

16. Evaluating performance advantages of grouping genetic algorithms (2004) (28)

Questo utilizza la stessa fitness function di 1;

17. Use of genetic algorithms in operations management Part 1: Applications (2004) (29)

$$\text{Fitness value} = L + 100N_a + 10N_b$$

dove  $L = [(N_w CT) - \sum_{i=1}^{i=n} t_i * 100](N_w CT)^{-1}$  rappresenta il *balancing loss* (%),  $N_w$  il numero totale di stazioni,  $CT$  il tempo ciclo,  $t_i$  il tempo di processazione della operazione *i-esima*,  $n$  il numero di operazioni,  $N_a$  il numero di precedenze e  $N_b$  il numero di stazioni che eccedono il tempo ciclo.

Questa funzione obiettivo che calcola la *balancing loss* per ciascuna soluzione e aggiunge una penalità per quelle che permettono alle stazioni di lavoro di eccedere il tempo ciclo e/o di violare le precedenze;

18. Network models and optimization: multi-objective genetic algorithm approach (2008) (30)

$$F(T_s(j)) = W_1 * (W_1 - F_1(T_s(j))) + W_2 * (F_2(T_s(j)) - Z_{2,min}) + W_3 * (1 - F_3(T_s(j)) - Z_{3,min}),$$

$$j = 1, \dots, Popsiz$$

dove:

- $F_1$ : minimizzazione del numero di stazioni
- $F_2$ : massimizzazione della efficienza di bilanciamento

- $F_3$ : minimizzazione della varianza del carico di lavoro fra le stazioni (Workload Variance)
- $T_s(j)$  è la sequenza di operazioni della soluzione corrente
- $W_1$  è impostato come un numero grande così da rendere  $F_1$  un obiettivo non dominato
- i pesi  $W_2$  e  $W_3$  sono calcolati come:

$$W_k = \frac{1}{Z_{k,max} - Z_{k,min}}, k = 2,3$$

dove:

- $Z_{k,max}$  è il valore massimo dell'obiettivo  $k$ -esimo nella popolazione della generazione corrente
- $Z_{k,min}$  è il valore minimo dell'obiettivo  $k$ -esimo nella popolazione della generazione corrente;

#### 19. Assembly line balancing based on an adaptive genetic algorithm (2009) (31)

$$f = S_{max} - \sqrt{\sum_{k=1}^n (C_{max} - T_k)^2} - 2n$$

dove  $n$  rappresenta il numero di stazioni,  $C_{max}$  il tempo ciclo desiderato,  $T_k$  il tempo di assemblaggio dell'operazione  $k$ -esima e  $S_{max}$  una costante (grande) che assicura che la funzione obiettivo sia sempre non negativa.

La prima parte della funzione ( $\sqrt{\sum_{k=1}^n (C_{max} - T_k)^2}$ ) (Smoothness Index) trova la migliore soluzione di bilanciamento con lo stesso numero di stazioni; la seconda parte minimizza il numero di stazioni. Il peso è 2 perché questa seconda parte viene considerata più importante della prima;

#### 20. Novel Approach for Balancing Manual Automobile Assembly Based on Genetic Algorithm (2010) (32)

$$\min WV = \frac{1}{N} \sum_{j=1}^N \left[ T_j - \left( \sum_{i=1}^I t_i \right) / N \right]^2$$

dove  $N$  rappresenta il numero di stazioni,  $T_j$  il carico della stazione  $j$ -esima e  $t_i$  il tempo di esecuzione dell'operazione  $i$ -esima;

21. Dynamic segregative genetic algorithm for assembly line balancing (2010) (33)

$$fit(x) = \omega_1 * (mC - T_{tot}) + \omega_2 \sqrt{\frac{1}{m} \sum_{j=1}^m (C - T(W_j))^2}$$

dove  $\omega_1$  e  $\omega_2$  rappresentano i pesi ( $\omega_1 + \omega_2$ ),  $m$  il numero di stazioni,  $C$  il tempo ciclo,  $T_{tot}$  il tempo totale necessario all'assemblaggio del prodotto e  $T(W_j)$  il tempo necessario all'operatore della stazione  $j$ -esima di completare l'operazione di montaggio.

Il primo termine rappresenta il tempo di attesa totale delle stazioni, e il secondo termine la distribuzione del carico di lavoro fra le stazioni (Workload Smoothness);

22. Biologically Inspired Genetic Algorithm to Minimize Idle Time of the Assembly Line Balancing (2011) (34)

$$Minimize T_{id} = \sum_{i=1}^m (CT - T_i)$$

dove  $CT$  rappresenta il tempo ciclo,  $m$  il numero di stazioni e  $T_i$  il carico della stazione  $i$ -esima;

23. A multi-objective approach based on genetic algorithm for multi-model line process planning considering difference in worker ability (2011) (35)

$$f = \sum_{i=1}^4 w_i f_i$$

dove  $\forall w_i = \frac{random_i}{\sum_{i=1}^4 random_i} \in (0,1), \sum_{i=1}^4 w_i = 1$

Le quattro funzioni obiettivo sono focalizzate su:

- il tempo ciclo ( $CT$ ) di ogni modello  $l$ :

$$\min f_1 = \sum_{l=1}^L CT_l$$

quando le dimensioni del lotto sono le stesse,

$$\min f_1' = \sum_{l=1}^L w_l CT_l, \quad w_l = LS_l$$

quando le dimensioni  $LS_l$  non sono le stesse.

- la varianza del carico di lavoro fra le stazioni (Workload Variance) di ogni modello  $l$ :

$$\min f_2 = \sqrt{\sum_{l=1}^L f_2'}, \quad \forall l, f_2' = \frac{1}{W} \sum_{ws=1}^W (CT_l - TT_{l,ws})^2$$

- il tempo di commutazione (Switch Time), ovvero il costo del tempo dovuto al cambiamento da un modello all'altro in ogni stazione di lavoro. Lo switch time fra il modello  $l$  e il modello  $p(l)$  è dovuto alla differenza fra la posizione assegnata ai lavoratori.

$$\min f_3 = \sum_{\forall l, p(l) > 0} SwitchTime_{p(l),l}$$

- il tempo di processazione totale:

$$\min f_4 = makespan;$$

24. A novel feasible task sequence-oriented discrete particle swarm algorithm for simple assembly line balancing problem of type 1 (2013) (36)

Questo utilizza la stessa fitness function di 12;

25. Development of a genetic algorithm for multi-objective assembly line balancing using multiple assignment approach (2014) (3)

Questo utilizza la stessa fitness function di 18.

Si riporta a questo punto la tabella riassuntiva (Tabella 3.1) dell'analisi effettuata sullo stato dell'arte delle fitness function, mettendo in evidenza l'obiettivo(i) della funzione e il problema al quale essa si applica.

N	Anno	Ricercatore(i)	Titolo Articolo	Obiettivo(i)										Problem Type	
				Max Workl Smooth	Min Num stazioni	Max Eff bilanc	Min Tempo ciclo	Min Switch Time	Min Tempo tot process	Min freq cambi attrezz	Min num attrezz	Max Workl Relatd	Max n obiett		
1	1991	Falkenauer, E.	A genetic algorithm for grouping	<input checked="" type="checkbox"/>											SALBP-1
2	1992	Falkenauer e Delchambre	A genetic algorithm for bin packing and line balancing	<input checked="" type="checkbox"/>											SALBP-1
3	1994	Leu Y. Y., Matheson, L.A., Rees, L.P.	Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Population and Multiple Evaluation Criteria	<input checked="" type="checkbox"/>											SALBP-1
4	1995	Yasuhiro Tsujimura, Mitsuo Gen e Erika Kubota	Solving fuzzy Assembly-line Balancing Problem with Genetic Algorithms	<input checked="" type="checkbox"/>											SALBP-1
5	1996	Tadahiko Murata, Hisao Ishibuchi e Hideo Tanaka	Multi-objective genetic algorithm and its applications to flowshop scheduling											<input checked="" type="checkbox"/>	SALBP-1
6	1996	Yeo Keun	Genetic	<input checked="" type="checkbox"/>									<input checked="" type="checkbox"/>		SALBP

		Kim, Yong Ju Kim e Yeongho Kim	Algorithms for assembly line balancing with various objectives											
7	1996	Falkenauer, E.	A hybrid grouping algorithm for bin packing	<input checked="" type="checkbox"/>										SALBP-1
8	1996	Mitsuo Gen, Yashuhiro Tsujimura e Yinxiu Li	Fuzzy Assembly line balancing using genetic algorithms	<input checked="" type="checkbox"/>										SALBP-1
9	1997	Falkenauer, E.	A grouping genetic algorithm for line balancing with resource dependent task times	<input checked="" type="checkbox"/>										SALBP-1
10	1998	Hisao Ishibuchi e Tadahiko Murata	A Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling									<input checked="" type="checkbox"/>		SALBP-1
11	1998	Yong Ju Kim Yeo Keun Kim e Yongkyun Cho	A Heuristic-based genetic algorithm for workload smoothing in assembly lines	<input checked="" type="checkbox"/>										ALBP
12	2000	Sabuncuoglu I., Erel E., Tanyer	Assembly line balancing using genetic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>									SALBP-1

		M.	algorithms												
13	2000	S.G. Ponnambalam P. Aravindan, G. Mogileeswarar Naidu	A multi-Objective Genetic Algorithm for Solving Assembly Line Balancing Problem											<input checked="" type="checkbox"/>	SALBP-1
14	2000	Joaquin Bautista Raul Suarez, Manuel Mateo, Ramon Companys	Local Search Heuristics for the Assembly Line Balancing Problem with Incompatibilities Between Tasks	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>										SALBP-1
15	2002	Ruey-Shun Chen Kun-Yung Lu, Shien-Chiang Yu	A hybrid genetic algorithm approach on multi-objective of assembly planning problem	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				SALBP-2
16	2004	Evelyn C., Brown Robert e T. Sumichrast	Evaluating performance advantages of grouping genetic algorithms	<input checked="" type="checkbox"/>											SALBP-1
17	2004	D J Stockton, L Quinn, R A Khalil	Use of genetic algorithms in operations management Part 1: Applications	<input checked="" type="checkbox"/>											SALBP
18	2008	Gen M Cheng R e Lin L.	Network models and optimization: multiobjective genetic algorithm	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>									SALBP

			approach											
19	2009	Yin Jianfeng Yu e Yuehong	Assembly line balancing based on an adaptive genetic algorithm	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>									SALBP-1
20	2010	Qihua Tang Zhonghua Xiao, Yanli Liang, Mingxing Deng e Zhongmin Xi	Novel Approach for Balancing Manual Automobile Assembly Based on Genetic Algorithm	<input checked="" type="checkbox"/>										SALBP
21	2010	Octav Brudaru Cristian Rotaru	Dinamic segregative genetic algorithm for assembly line balancing	<input checked="" type="checkbox"/>										SALBP-1
22	2011	Noraini Mohd Razali e John Geraghty	Biologically Inspired Genetic Algrithm to Minimize Idle Time of the Assembly Line Balancing	<input checked="" type="checkbox"/>										ALBP
23	2011	Jiahua Weng, Xianchao Wu, Hisashi Onari	A multi-objective approach based on genetic algorithm for multi-model line process planning considering difference in worker abilit	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					ALBP
24	2013	Su	A novel feasible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>									SALBP-1

		Jianping Dou & Jun Li e Chun	task sequence-oriented discrete particle swarm algorithm for simple assembly line balancing problem of type 1											
25	2014	Tarek Al-Hawari Marwan Ali, Omar Al-Araidah, Ahmad Mumani	Development of a genetic algorithm for multi-objective assembly line balancing using multiple assignment approach	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								SALBP

Tabella 3.1: Analisi dello stato dell'arte delle fitness function e degli obiettivi utilizzati per la risoluzione dei problemi di ottimizzazione dell'assemblaggio

### 3.3.1 Risultato delle analisi

Per concludere l'analisi svolta riguardo a come in letteratura si differenziano e applicano le fitness function degli algoritmi genetici per risolvere i problemi di ottimizzazione dell'assemblaggio, si è realizzato un diagramma a torta (Figura 3.7) che mostra le percentuali di utilizzo dei vari obiettivi citati nel precedente paragrafo.

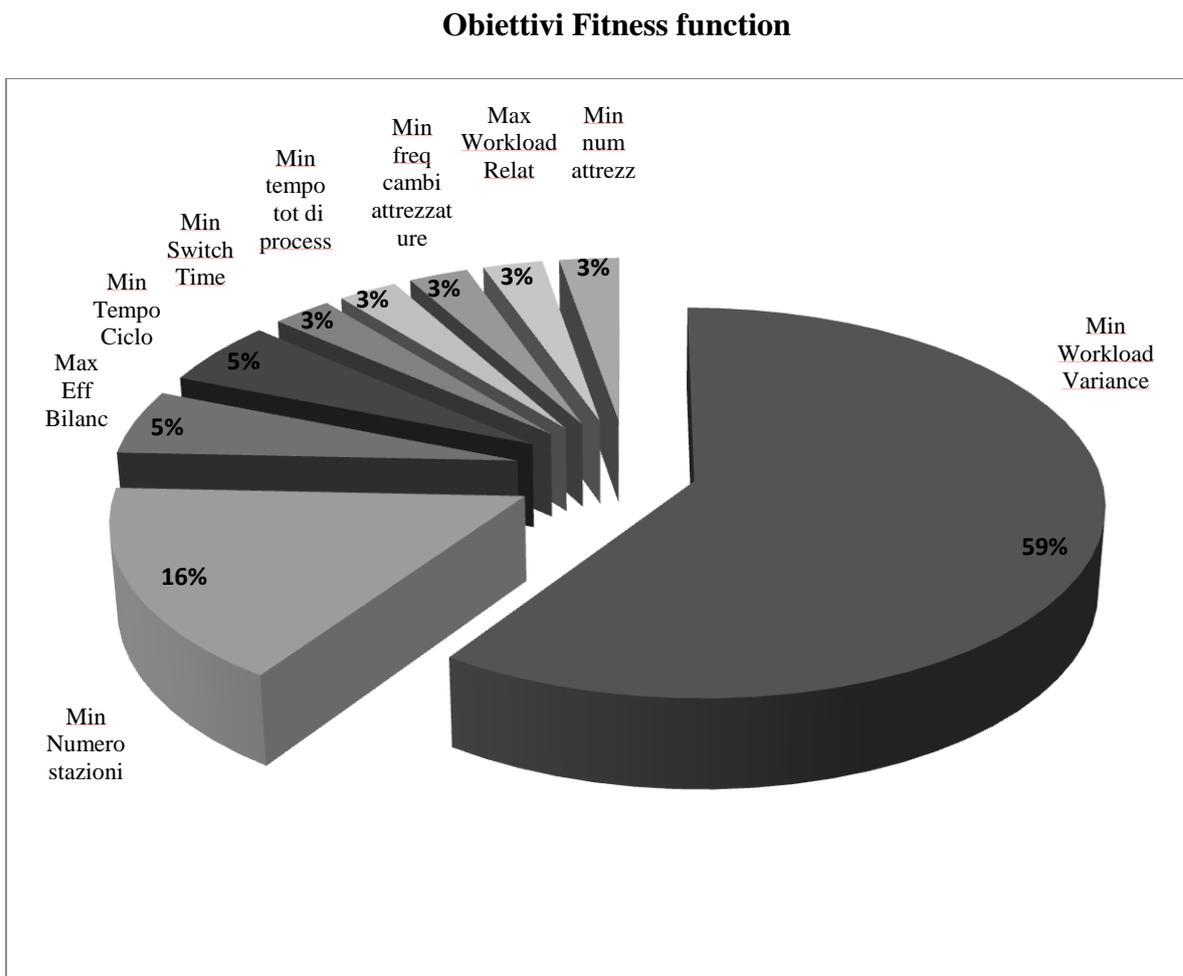


Figura 3.7: Diagramma a torta con percentuali di utilizzo degli obiettivi delle fitness function

Come si nota dal diagramma, gli obiettivi di maggiore impiego sono: la massimizzazione della distribuzione del carico di lavoro (Workload Smothness), la minimizzazione del numero di stazioni, la massimizzazione dell'efficienza di bilanciamento e la minimizzazione del Tempo Ciclo. Questi, o in funzioni mono-obiettivo, o associati fra di loro, o ad altri in funzioni multi-obiettivo, sono gli obiettivi tipicamente presi in considerazione. Gli altri sono studiati per una o al massimo un paio di applicazioni, ma non

per questo ritenuti meno critici dal punto di vista strategico dell'efficienza di una linea di montaggio manuale.

### **3.4 Considerazioni conclusive sull'analisi dello stato dell'arte**

Alla luce di quanto descritto in questo terzo capitolo si evidenzia per prima la caratteristica fondamentale dell'algoritmo genetico nella risoluzione di problemi di ottimizzazione del montaggio, ovvero la sua capacità di convergere verso soluzioni sempre migliori senza dover necessariamente esaminare ogni possibile soluzione del problema, con tempi computazionali relativamente brevi (a differenza delle tecniche tradizionali sia euristiche che iterative)

L'algoritmo genetico è uno strumento molto flessibile, non solo per i vari campi di applicazione, ma anche perché concede per ogni suo passo varie modalità di esecuzione. Questa caratteristica permette all'utilizzatore di poter migliorare la soluzione provando a cambiare la tipologia di uno o più step e vedere come essa reagisce, con particolare riferimento alla rappresentazione genetica, alla selezione e agli operatori genetici di crossover e mutazione.

L'elemento che più è in grado di distinguere un algoritmo da un altro e che più sprona i ricercatori nello studio approfondito di questa tecnica è la fitness function: essa concede la possibilità di decidere quali sono gli obiettivi da raggiungere e, costituendo fattore di bontà della soluzione, funge da punto di forza dell'algoritmo.

A valle dello studio effettuato nel precedente paragrafo sulla letteratura presente riguardo le fitness function, le considerazioni evidenti sono:

- più del 50 % delle funzioni sono mono-obiettivo;
- di questo 50 %, più della metà hanno come obiettivo la massimizzazione della distribuzione del carico di lavoro fra le stazioni;
- l'altra metà è tipicamente associata alla minimizzazione del numero di stazioni o alla massimizzazione dell'efficienza di bilanciamento in funzioni proposte a due obiettivi;

Si evince perciò come i sistemi sviluppati fino ad ora siano essenzialmente concentrati su obiettivi generici se pur importanti, trascurando però aspetti caratteristici e strategici per un efficiente bilanciamento di una linea manuale. Si è quindi deciso di sviluppare un algoritmo genetico multi obiettivo in grado di ottimizzare simultaneamente: il numero di stazioni della linea, la Workload Variance, il numero di operatori aventi competenze tecniche elevate, il numero di equipment utilizzati sulla linea e il numero di cambi di direzione necessari al completo assemblaggio del prodotto. Il tutto nell'ottica di rendere economicamente più efficiente l'intero impianto.

## Capitolo 4 Motivazioni e obiettivi della tesi

Prima di inoltrarci sulle caratteristiche centrali ed innovative dell'algoritmo, e quindi prima di descrivere gli obiettivi proposti, è necessario definire la tipologia di linea di produzione cui facciamo riferimento, proprio per comprendere il perché delle scelte che faremo.

Le assunzioni principali che si fanno sulla linea (già descritte in modo generico nel capitolo 2) sono le seguenti:

- produzione di massa;
- linea seriale unilaterale;
- linea single-model;
- tempi deterministici;
- intervallo di lancio fisso.

### 4.1 Obiettivi proposti per valutare la funzione dell'algoritmo genetico

Gli obiettivi proposti sono i seguenti:

- minimizzazione del numero delle stazioni;
- minimizzazione delle abilità degli operatori fra le stazioni;
- minimizzazione degli equipment fra le stazioni;
- minimizzazione dei cambi di orientamento dei componenti in una certa sequenza;
- massimizzazione del livellamento del carico fra le stazioni ovvero la minimizzazione della Workload Variance.

Secondo la ricerca illustrata nel capitolo 3 riguardo l'analisi dello stato dell'arte delle funzioni obiettivo, solo il primo e l'ultimo obiettivo sono stati utilizzati negli algoritmi analizzati (minimizzazione del numero di stazioni e massimizzazione del livellamento del carico).

Inoltre, a meno del primo parametro che fa parte per definizione del problema SALBP-1 cui facciamo riferimento, e che quindi è d'obbligo raggiungere, gli altri provengono da considerazioni ricercate e motivate che si descrivono nei prossimi paragrafi.

#### **4.1.1 Minimizzazione del numero di stazioni di lavoro**

Come già visto nel precedente capitolo, la minimizzazione del numero di stazioni rappresenta uno degli obiettivi più utilizzati nelle funzioni di fitness.

I vantaggi che si traggono dall'ottimizzazione del presente parametro derivano dal fatto che nel momento in cui si installa una nuova linea di montaggio, un numero inferiore di stazioni comporta:

- minori costi di mano d'opera;
- minori costi di attrezzature e macchinari;
- minori costi di set-up e manutenzione;
- maggior saturazione delle stazioni installate, e quindi maggiore livellamento del carico di lavoro;
- minori tempi.

#### **4.1.1 Minimizzazione delle abilità degli operatori fra le stazioni**

La gestione efficace della manodopera rappresenta una delle funzioni più critiche interne all'azienda, in grado di influenzare le prestazioni dei sistemi produttivi; questo è vero soprattutto nei settori dove il costo del lavoro è notevolmente alto a causa della presenza di operazioni particolarmente delicate che richiedono manodopera molto qualificata. Le aziende sono continuamente alla ricerca di soluzioni in grado di gestire ed ottimizzare il sistema, con particolare riferimento alla possibilità di trarre vantaggio dalle varie qualità presenti.



Figura 4.1: Operatore specializzato nell'utilizzo di un avvitatore

L'obiettivo proposto tiene conto del fatto che non tutti gli operatori hanno le stesse capacità, abilità e competenze rispetto a specifiche mansioni. I requisiti che infatti qualificano un lavoratore sono determinati dall'operazione più complessa che questo può essere in grado di svolgere. Si mostra in Figura 4.1 un operatore specializzato nell'utilizzo di un avvitatore elettrico, oggetto che, per quel tipo di attività, determina in gran parte le prestazioni finali del prodotto e considerato complesso da maneggiare, perciò assegnato a personale qualificato.

Per questo motivo può essere necessario concentrare operazioni complicate in poche stazioni nelle quali si posizionano tali operatori. Il corretto posizionamento permette di ridurre gli sprechi (in termini di ore di lavoro) nel processo produttivo.

Raggiungere questo obiettivo ha un principale riscontro positivo sul processo produttivo:

- sostanziale riduzione dei tempi improduttivi: un operatore poco abile potrebbe impiegare più tempo del dovuto nell'eseguire una certa operazione complessa o addirittura non riuscire a portarla a termine. Questi inconvenienti a loro volta possono creare problemi di movimentazione del prodotto lungo la linea: dal momento che stiamo parlando di sistemi a cadenza fissa, le inabilità degli operatori possono condurre al mancato completamento del lavoro sulla stazione nell'intervallo di tempo concesso. Tra l'altro questi sistemi possono creare ansia e

stress agli operatori che sono obbligati ad eseguire le operazioni entro le tempistiche date; a maggior ragione quindi le operazioni più complesse devono essere disposte fra gli operatori più abili.

A titolo di esempio si cita il caso Fendi (37).

In una pelletteria dove si lavora principalmente in modo artigianale e destrutturato, la componente umana e dunque le relative competenze dei singoli, sono fondamentali. Gli operatori sono altamente specializzati e per poter realizzare ogni tipo di operazione è essenziale la componente esperienza.

Nel contesto di riferimento (Fendi), la problematica più grave riscontrata è dovuta al fatto che il carico di lavoro non è assegnato in base ai criteri di ottimizzazione, ma la divisione è dettata dalle esigenze del momento, lasciando agli operatori stessi la facoltà di decidere come meglio organizzarsi. Questa mancanza di organizzazione comporta:

- inefficienza produttiva;
- sprechi di tempo in quanto tra la fine di una fase lavorativa e la successiva intercorre un periodo di tempo in cui l'operatore deve decidere quali siano le priorità di lavorazione e organizzarsi insieme agli altri;
- possibilità di attribuire ad un operatore un'operazione per la quale non è altamente specializzato nonostante siano presenti altri che invece sono più abili nella sua realizzazione.

Più in dettaglio, il reparto interno Fendi presenta le seguenti caratteristiche:

- numero di operatori limitato;
- competenze altamente specializzate e non completamente intercambiabili;
- assegnazione dei compiti non prestabilita e strutturata, ma lasciata alle esigenze del momento.

Quello che Fendi ha quindi cercato di fare, sulla base di queste problematiche evidenziate, è di riorganizzare al meglio il percorso all'interno del reparto produttivo, bilanciando il carico di lavoro in modo che tutti lavorassero in maniera continua ed equilibrata, senza trovarsi nella situazione di non sapere cosa fare e attribuendo mansioni delicate e complesse agli operatori più abili.

Inoltre nell'approccio utilizzato dalla tesi in riferimento all'assegnazione delle operazioni, è considerato di fondamentale importanza conoscere le persone che si hanno a disposizione e le competenze e caratteristiche di ciascuna di esse.

Tali considerazioni fanno capo alla decisione di utilizzare la minimizzazione delle abilità degli operatori fra le stazioni come parametro di linea dell'algoritmo genetico.

#### 4.1.2 Minimizzazione degli equipaggiamenti fra le stazioni

Con il termine equipment (attrezzo) si intende uno strumento meccanico e non, adoperato per eseguire una determinata lavorazione manuale; alcuni attrezzi possono essere utilizzati singolarmente, mentre altri hanno bisogno di essere accoppiati fra di loro per poter svolgere correttamente la propria funzione.



Figura 4.2: Alcuni degli attrezzi generalmente utilizzati nelle linee di montaggio manuali

In Figura 4.2 sono illustrati alcuni dei molteplici utensili che si possono utilizzare manualmente per eseguire operazioni meccaniche e nelle Figure 4.3 e 4.4 esempi reali di operatori che adoperano tali equipment. Tra i più comuni: alesatori, cacciavite, forbici e cesoie, martelli, pinze, presse, punte da trapano, trapani e tutti gli strumenti di misura. Oltre a questi, fanno parte degli equipment anche i macchinari che permettono di eseguire determinate operazioni, in alcuni casi anche tipiche dell'attività svolta o del settore di impiego (ad esempio una cucitrice, tipica del settore tessile).



Figure 4.3 e 4.4: Esempi di equipment utilizzati in stazioni di lavoro manuali

Lo scopo di questo parametro è quello di minimizzare gli equipment da utilizzare in ogni stazione di lavoro situata sulla linea di montaggio e determinare come assegnare le operazioni nelle varie stazioni, con l'obiettivo di ridurre il numero di ciascun attrezzo.

Minimizzare gli equipment fra le stazioni significa perciò cercare di concentrare le operazioni che necessitano degli stessi utensili o macchinari nelle medesime stazioni di lavoro, così da ridurre il numero e di conseguenza i costi ad essi associati. In particolare, quando si parla di equipment tipici dell'attività svolta dall'azienda, si tratta nella maggior parte dei casi di strumenti sofisticati che hanno costi elevati sia di acquisto che di manutenzione e dei quali non si può fare a meno per la finalizzazione e la buona riuscita del prodotto.

Si illustra in Figura 4.5 un esempio di strumento tipico di un'attività: una macchina scarnatrice di pelli caratteristica dell'industria tessile.



Figura 4.5: Macchina per la scarnitura delle pelli

Diventa opportuno perciò tener conto di alcuni aspetti nel bilanciamento della linea:

- più operazioni possono necessitare degli stessi equipment, eventualmente anche nello stesso momento o comunque per un periodo in concomitanza;
- se un attrezzo è particolarmente costoso e difficile da utilizzare, potrebbe risultare vantaggioso affidarlo ad un operatore competente in un'unica stazione di lavoro.

A titolo di esempio si cita sempre il caso Fendi (37).

Si riporta il caso delle macchine da cucire. Alcune operazioni, dette “operazioni da macchina”, prevedono l'utilizzo di macchine da cucire molto costose in grado di effettuare cuciture su pelle e fodera con elevatissime prestazioni; tali prestazioni però, dipendono anche dalla capacità delle macchiniste.

Nell'approccio utilizzato dalla tesista, per ovviare a questo problema, si concentra la macchina da cucire in un'unica stazione di lavoro: questo permette di minimizzare i costi della macchina stessa e di fare in modo che le operazioni su di essa siano effettuate da macchiniste abili. Concentrare le macchiniste abili nella stazione in cui è posizionata la

cucitrice permette di ridurre i tempi improduttivi causati dalla incapacità e poca esperienza degli altri operatori.

È evidente, e questo esempio ne è una chiara dimostrazione, che la correlazione fra questo obiettivo e quello descritto prima (minimizzazione delle abilità degli operatori fra le stazioni) è molto stretta; a maggior ragione si ritiene quindi necessaria la presenza di entrambi i parametri nella valutazione della funzione obiettivo dell'algoritmo implementato.

#### **4.1.3 Minimizzazione dei cambi di orientamento dei componenti in una certa sequenza**

Ogni sequenza di montaggio prevede lo svolgimento di una serie di operazioni sui componenti che portano all'assemblaggio di un prodotto; il montaggio di un assieme in una stazione manuale è caratterizzato da due fasi principali:

- fase di manipolazione;
- fase di assemblaggio vero e proprio (con o senza equipment).

In particolare, la fase di manipolazione a sua volta prevede:

- la presa del componente dal luogo dove è immagazzinato;
- il corretto orientamento e avvicinamento all'area di assemblaggio.

Quando si parla di corretto orientamento ci si riferisce alla fase in cui si colloca il componente nella posizione tale da predisporlo per essere montato. Per questa fase e per la successiva di assemblaggio vero e proprio si fanno alcune ipotesi:

- si ipotizza (semplificando), che le operazioni svolte sui vari componenti si realizzino sulle tre direzioni, in senso positivo o negativo ( $\pm x$ ,  $\pm y$ ,  $\pm z$ );
- si ipotizza inoltre che i componenti vengano assemblati uno alla volta con una sequenza che nella maggior parte dei casi va dall'alto verso il basso con asse di inserimento in direzione rettilinea verticale.

Lo scopo di questo parametro è quello di cercare di minimizzare i cambi di orientamento necessari alla sequenza di assemblaggio per realizzare il prodotto finito.

Quando si parla di rotazione del pezzo, si fa riferimento sia alla rotazione manuale che a quella eseguita con l'ausilio di qualche apparecchio.

Nelle Figure 4.6 e 4.7 si presentano due tavole di montaggio per il sollevamento, l'orientamento e la rotazione di componenti da 250 kg (la prima) fino a 500 kg (la seconda).

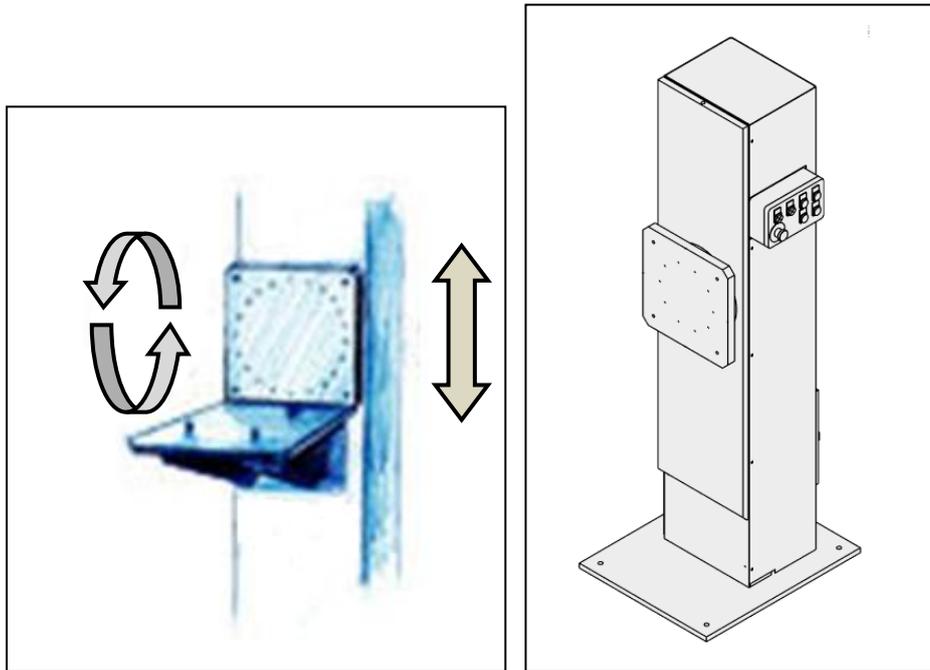


Figure 4.6 e 4.7: Tavole di montaggio

L'obiettivo in entrambi i casi rimane invariato, e si può raggiungere cercando di evitare rotazioni e cambi di direzione continui (della tavola o del pezzo se montato a mano), ma associando tra loro le operazioni che si svolgono lungo una medesima direzione.

Il vantaggio principale raggiungibile con questo parametro è la riduzione dei tempi di assemblaggio e di conseguenza la riduzione dei tempi produttivi.

#### 4.1.4 Massimizzazione del livellamento del carico di lavoro fra le stazioni

Questo parametro, dall'analisi dello stato dell'arte del Capitolo 3, si ricorda essere quello più utilizzato nelle funzioni individuate.

Con varianza del carico di lavoro si fa riferimento al livellamento della distribuzione dello stesso fra le stazioni. In Figura 4.8 si illustra un semplice esempio di bilanciamento di una linea di assemblaggio dal quale si evidenzia la presenza di un dislivello fra le varie stazioni di lavoro: a parte l'ultima stazione, tutte le altre mostrano elevati tempi di attesa.

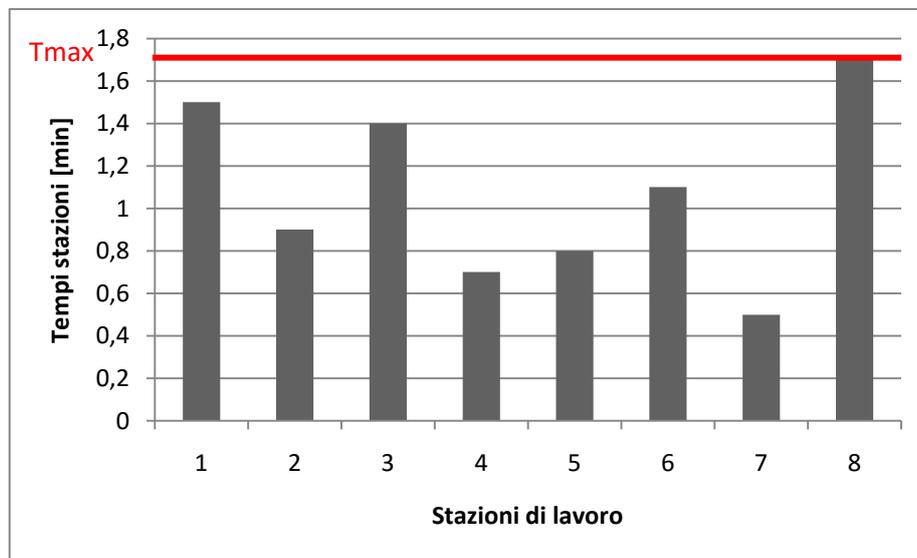


Figura 4.8: Esempio di bilanciamento di una linea di assemblaggio

Lo scopo di questo obiettivo è quello di rendere il lavoro fra le stazioni il più equo possibile, facendo sì che il concetto di migliore distribuzione possa condurre a vantaggi quali:

- aumento del rateo produttivo;
- mitigazione delle differenze di carico di lavoro fra gli operatori.

In letteratura sono presenti svariati modi di valutare il livellamento del carico sia in termini di massimizzazione della Workload Smoothness, sia di minimizzazione della Workload Variance; in questa tesi si parla in termini di minimizzazione di  $WV$  e la formula che si utilizza sarà descritta nel prossimo paragrafo.

## 4.2 Caratteristiche dell'algoritmo genetico implementato

Nel capitolo 3 si sono descritti i cinque passi principali di un algoritmo genetico definendo tutte le modalità con cui ciascuno di essi può essere eseguito. Si definiscono a questo punto nel dettaglio gli step dell'algoritmo implementato e le caratteristiche di ognuno; prima però, si elencano quali sono i dati in input che lo strumento deve avere a disposizione:

- numero di operazioni elementari necessarie per assemblare il prodotto;
- rateo produttivo necessario a soddisfare la domanda annua [pezzi/ora];
- tempo di riposizionamento [min];
- efficienza della linea;
- per ciascuna operazione elementare si deve a sua volta conoscere:
  - il tempo di esecuzione dell'operazione [min];
  - le precedenze tecnologiche;
  - la skill richiesta;
  - l'eventuale bisogno di equipment(s), e se sì la definizione di quale/quali;
  - la direzione di assemblaggio;
- numero di iterazioni dell'algoritmo;
- numero di individui (cromosomi);
- frazione della popolazione (parte della popolazione sottoposta a riproduzione);
- probabilità dell'operatore genetico crossover;
- probabilità dell'operatore genetico mutazione.

### 1 Rappresentazione genetica e generazione iniziale della popolazione iniziale di soluzioni

La strategia utilizzata nel presente lavoro per costruire il set di operazioni (geni) di una soluzione è la “*task-oriented representation*”. Affinché la soluzione sia in grado di mantenere la fattibilità si introduce il seguente “*sequence planner*”:

1. si crea un set di operazioni utilizzabili ovvero prive di precedenze tecnologiche;
2. se il set è un insieme vuoto allora la procedura termina, altrimenti step 3;

3. si sceglie random una tra queste operazioni, si immette nella prima posizione disponibile del cromosoma e si toglie dal set di operazioni utilizzabili;
4. si inseriscono nel set di operazioni utilizzabili quelle che succedono l'operazione rimossa se non vincolate da operazioni ancora non eseguite;
5. si reitera dallo step 2.

Notare che nello step 4 il set disponibile viene aggiornato con operazioni che soddisfano le precedenze e questo assicura la generazione di una soluzione fattibile.

## 2 Valutazione della Funzione di Fitness per le soluzioni che appartengono all'attuale popolazione

La funzione di fitness si calcola come:

$$F = \eta_1 F_1 + \eta_2 F_2 + \eta_3 F_3 + \eta_4 F_4 + \eta_5 F_5$$

Dove:

- $F_i$  = valori non normalizzati;
- $F_i = 0 \div 1$ , valori normalizzati; si sceglie di normalizzare i valori degli obiettivi per fare in modo che la funzione possa risultare equa per ciascuno di essi;
- $\eta_i$  = pesi,  $\sum \eta_i = 1$ ;
- $F_i = 1 + (LB - x)/(UB - LB)$ , dove  $x$  rappresenta la variabile da normalizzare in questione;
- $F_1$  = parametro normalizzato che rappresenta la minimizzazione del numero di stazioni ( $Min N$ );
- $F_2$  = parametro normalizzato che rappresenta la minimizzazione delle abilità degli operatori fra le stazioni ( $Min S = \sum S_i \quad i = 1, \dots, n$  con  $n$  = numero di stazioni);
- $F_3$  = parametro normalizzato che rappresenta la minimizzazione degli equipaggiamenti fra le stazioni ( $Min E = \sum E_i \quad i = 1, \dots, n$ );
- $F_4$  = parametro che rappresenta la minimizzazione dei cambi di orientamento nella sequenza  $D$  ( $Min D = \sum D_i \quad i = 1, \dots, k$  dove  $k$  = operazioni);

- $F_5$  = parametro che rappresenta la minimizzazione del livellamento del carico fra le stazioni, ovvero la minimizzazione della Workload Variance ( $Min WV = \sum (T_{ai} - T_{am})^2 / N \quad i = 1, \dots, n$ ).

$LB$  e  $UB$  sono rispettivamente il limite inferiore e superiore (lower, upper bound) di ciascun obiettivo; di seguito, nella Tabella 4.1, si mostra come si determinano in questo lavoro.

Parametro obiettivo	X	LB	UB
<b>Numero stazioni</b>	N Totale stazioni	$Int T_p / T_c$	● <sub>1</sub>
<b>Abilità (skill) operatori</b> 0 = nessuna abilità 1 = abilità intermedia 2 = elevata abilità	S Somma delle abilità tra le stazioni	Max tra tutte le abilità presenti nella sequenza di montaggio	● <sub>2</sub>
<b>Equipment</b> 1 = equip n°1 2 = equip n°2 ...	E Somma degli equipment tra le stazioni	$\sum$ Equipment necessari per il montaggio	● <sub>3</sub>
<b>Cambi di orientamento</b> $\pm x$ $\pm y$ $\pm z$	D Somma di cambi di orientamento nella sequenza	$\sum$ Direzioni presenti nella sequenza -1	● <sub>4</sub>
<b>Livellamento del carico di lavoro</b>	WV Somma delle attese tra le stazioni	0	● <sub>5</sub>

Tabella 4.1:  $LB$  e  $UB$  dei parametri obiettivo

Nel caso di  $LB$  ovvero di limite inferiore, con questo si intende il migliore dei casi che si può presentare per ciascun obiettivo; al contrario, l' $UB$  fa riferimento al caso peggiore che si può presentare per ogni obiettivo in quanto limite superiore. Si descrivono quindi le motivazioni della loro scelta.

Per quanto riguarda il numero totale di stazioni, esistono vari metodi per calcolare i limiti; per il limite inferiore si utilizza il metodo più logico e matematico (come si osserva dalla formula). Il limite superiore proviene invece dallo studio del comportamento di una serie di casi (presenti in letteratura), ed in particolare alla loro analisi al variare del  $LB$ , che ha portato alla definizione del seguente algoritmo, nel quale si assume che l' $UB$  di  $N$  sia dipendente dal  $LB$ :

●<sub>1</sub>  $UB$  del numero totale di stazioni:

1. Calcolare  $LB$
2.  $UB_n = Int(LB_n + xLB_n)$  where
3. If  $LB < 10$

*then  $x = 30\%$*

*Else if  $LB > 10$*

*and  $LB < 30$*

*then  $x = 20\%$*

*Else  $x = 10\%$*

*End if.*

La definizione dei limiti per i parametri che sono differenti dal numero totale di operazioni non può invece tener conto di quanto esiste in letteratura in quanto non si sono individuate funzioni che li utilizzassero o meglio, anche se si sono trovate, i parametri non sono stati normalizzati ed inseriti in una funzione multi-obiettivo pesata. Si sono perciò scelti dei limiti in modo arbitrario, seppur basandoci sulla logica e sulla pratica.

●<sub>2</sub> *UB* delle abilità degli operatori:

Sommatoria delle massime abilità tra le stazioni considerando come numero di stazioni l'*UB* di *N*.

●<sub>3</sub> *UB* degli equipment:

Come per *S*, si prende *UB* di *N* e si inseriscono gli equipment in modo tale che non si ripetano all'interno delle stazioni.

●<sub>4</sub> *UB* dei cambi di orientamento:

1. Calcolare *LB*

2. *If* { *LB* = 0      *then*    *UB* = 0 }

*Else if* { *LB* = 1      *then*    *UB* = 2*x* }

dove *x* = numero di volte in cui compare la direzione meno presente

*Else if*

{*for* { *LB* = 2 to *LB* = 5 }

*If* { *n<sub>i</sub>* > *k*/2    *i* = *x, y, z, -x, -y, -z* *then* *UB* = 2*x* }

where *x* = *k* - *n<sub>i</sub>* and *n<sub>i</sub>* = numero di volte in cui compare la direzione

*else*  $UB = k - 1$  }  $k =$  numero di operazioni.

●<sub>5</sub>  $UB$  della Workload Variance:

Si riempiono le stazioni (senza rispettare le precedenze tecnologiche) fino a che non si ha che  $T_{si} = T_{max}$  per ciascuna stazione, finché non viene raggiunto il  $T_p$ . Le rimanenti stazioni (il totale de numero di stazioni deve essere uguale all' $UB$ ) avranno  $T_{si} = 0$ .

Per questa configurazione si calcola:

$$UB_{WV} = \sum (T_{ai} - T_{am})^2 / N$$

### 3 Processo di riproduzione attraverso la selezione e gli operatori genetici di crossover e mutazione

In questo lavoro si utilizza:

- selezione: “*Roulette wheel selection*” (3.2.3);
- crossover: “*Uniform order based representation*”; come già descritto, questo operatore genetico considera due soluzioni adatte all'evoluzione e taglia i loro vettori di codifica in due punti predefiniti o casuali al fine di ottenere una testa, una parte centrale ed una coda dalla prima e dalla seconda soluzione. La prima nuova soluzione sarà data dalla testa e della coda della prima soluzione e dalla parte centrale formata dai geni mancanti così come appaiono nella seconda soluzione. La seconda nuova soluzione sarà data dalla testa e dalla coda della seconda soluzione e dalla parte centrale dai geni mancanti così come appaiono nella prima soluzione. Questa forma di crossover consente di generare cromosomi esclusivamente fattibili.

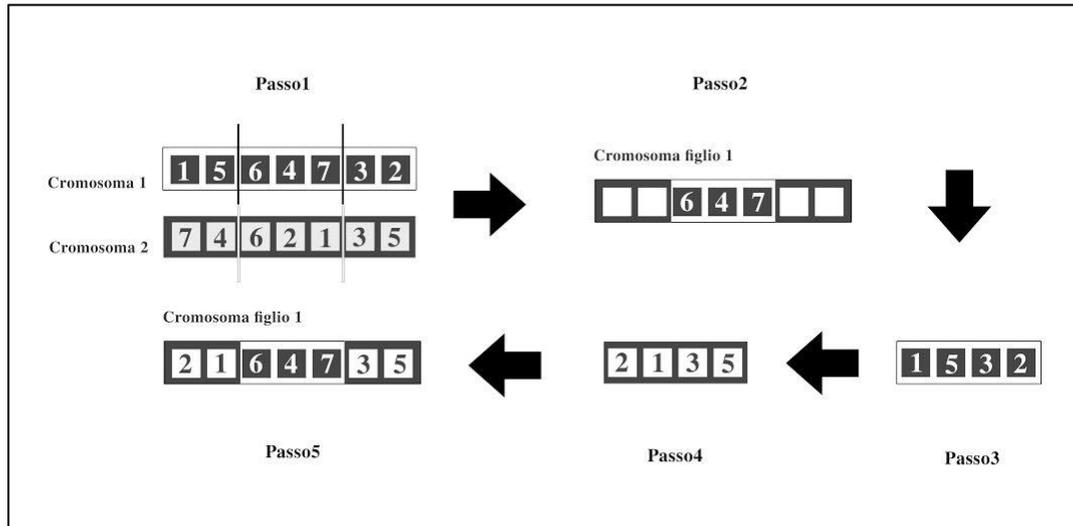


Figura 4.9: “Uniform order based crossover”

- mutazione: semplice ricombinazione casuale dei geni di uno stesso cromosoma. Si estraggono casualmente due numeri, e alle corrispondenti posizioni all’interno del cromosoma i geni vengono invertiti fra di loro.

#### 4 Riparazione dei cromosomi non fattibili ed inserimento dei nuovi cromosomi nella popolazione

La procedura del metodo di riparazione è la stessa usata per generare la popolazione iniziale (“*sequence planner*”), a differenza dello step 3 che si modifica con:

3. fra le operazioni disponibili all’interno del set, si seleziona quella che si trova nella prima posizione nel cromosoma infattibile e si aggiunge alla stringa.

Ovviamente essa va applicata alle soluzioni che nascono dall’operatore genetico di mutazione in quanto con l’*“order based crossover”* nascono soluzioni già fattibili.

## 5 Selezione e procedura finale

Dalla popolazione ottenuta (genitori più figli) vengono selezionati i cromosomi con i più alti valori della Fitness, i quali andranno a comporre la nuova generazione.

Ad ogni iterazione dell'algoritmo la numerosità della popolazione rimane invariata.

Per evitare la possibilità di perdere i migliori cromosomi come risultato della selezione della probabilità, si utilizza il metodo dell'"*elitismo*" descritto nel precedente capitolo.

Per quanto riguarda i criteri di stop applicati, nell'algoritmo implementato si utilizza un predeterminato numero di iterazioni che l'algoritmo deve eseguire e la ricerca si arresta quando viene raggiunto tale numero.

## Capitolo 5 Implementazione e validazione

Matlab (abbreviazione di Matrix Laboratory) è un ambiente per il calcolo numerico che consente di manipolare matrici, visualizzare funzioni e dati, implementare algoritmi, creare interfacce utente ed interfacciarsi con altri programmi.

Matlab è un software utilizzato da milioni di persone, sia nell'industria che nell'università, proprio grazie ai suoi numerosi strumenti a supporto dei più svariati campi di studio.

Nel presente lavoro si utilizza l'ambiente Matlab come intero supporto per la realizzazione dell'algoritmo implementato, in particolare:

- per la programmazione di tutti gli step dell'algoritmo (descritti nel paragrafo 4.2) attraverso la creazione di una serie di script (sequenza di comandi con estensione *.m*);
- per la creazione di una serie di interfacce di ausilio all'utente nell'inserimento dei dati in input di cui ha bisogno l'algoritmo e nella visualizzazione in output dei risultati ottenuti.

Le interfacce e la programmazione dei comandi per eseguire l'algoritmo genetico sono complementari fra di loro, ovvero:

- le interfacce di input forniscono i dati necessari ad alimentare i comandi di esecuzione dell'algoritmo i quali, una volta svolti, forniscono i risultati;
- tali risultati sono a loro volta necessari come input per le interfacce finali che permettono una visualizzazione chiara e completa della soluzione trovata dall'algoritmo.

**GenIAL** (**Genetic Iteration for Assembly Line**) è il nome dell'algoritmo genetico sviluppato e implementato in ambiente Matlab nel presente lavoro di tesi.

Si descrivono di seguito le interfacce create.

## 5.1 Interfacce grafiche

Nella realizzazione di un programma l'interfaccia grafica collegata al codice è di estrema utilità affinché un qualsiasi utente possa utilizzare il programma stesso.

Le interfacce utilizzate per l'algoritmo implementato sono molteplici e si espongono una ad una nei prossimi paragrafi:

- Interfaccia iniziale dove l'utente può specificare se il prodotto cui fa riferimento è nuovo o se è già stato inserito precedentemente;
- Interfaccia in cui si inseriscono i parametri del prodotto;
- Interfaccia in cui si inseriscono le caratteristiche delle operazioni elementari;
- Interfaccia in cui si inseriscono i parametri dell'algoritmo;
- Interfaccia riassuntiva di tutti i dati immessi;
- Due interfacce in cui si visualizza l'output del risultato.

### 5.1.1 Interfaccia grafica iniziale

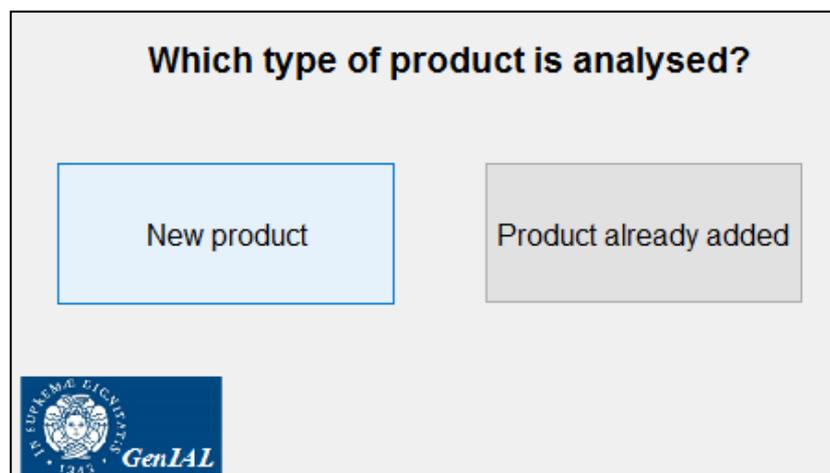


Figura 5.1: Interfaccia grafica iniziale

Attraverso questa interfaccia (Figura 5.1) l'utente può fare una prima distinzione fra l'analisi di un prodotto nuovo o uno che ha già studiato in precedenza. Se il prodotto è nuovo significa che non sono ancora noti i suoi parametri e tutte le informazioni circa le operazioni elementari. Se invece il prodotto è già esistente questo significa che l'utente ha

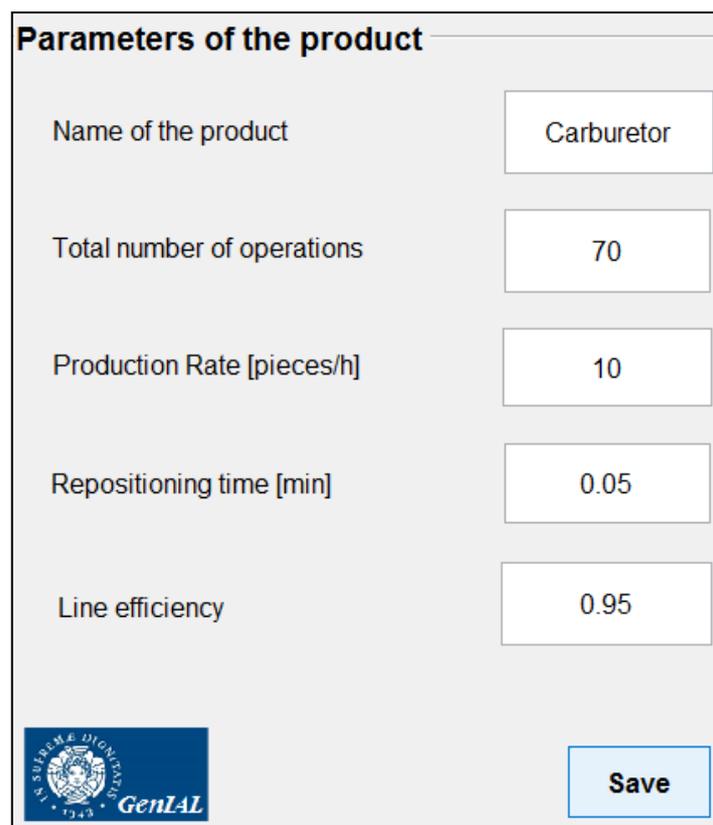
inserito e salvato i parametri del prodotto precedentemente e quindi può saltare la fase di immissione delle caratteristiche delle operazioni e passare direttamente all'introduzione dei parametri dell'algoritmo.

Con riferimento al caso di "prodotto nuovo" il passo successivo è l'interfaccia che si descrive nel prossimo paragrafo.

Se il prodotto è già esistente invece l'utente può accedere al database dove contenente tutti i prodotti già analizzati.

### 5.1.2 Interfaccia grafica del prodotto

A questo punto si chiede all'utente di inserire i dati circa il prodotto ed in particolare (come si mostra in Figura 5.2):



Parameters of the product	
Name of the product	Carburetor
Total number of operations	70
Production Rate [pieces/h]	10
Repositioning time [min]	0.05
Line efficiency	0.95

Save

GenLAL

Figura 5.2: Interfaccia grafica del prodotto

- il nome del prodotto;
- il numero totale di operazioni elementari;
- il rateo produttivo [pezzi/h];

- il tempo di riposizionamento [min];
- l'efficienza della linea.

Una volta introdotte queste informazioni, si passa all'inserimento delle caratteristiche di ciascuna operazione.

### 5.1.3 Interfaccia grafica delle operazioni elementari

Attraverso la presente interfaccia grafica l'utente immette tutte le operazioni elementari necessarie all'assemblaggio del prodotto ed in particolare, per ognuna di esse, informazioni circa (Figura 5.3):

**Carburetor** **Operation 21**

Precedences	
9	<input type="checkbox"/>
10	<input type="checkbox"/>
11	<input type="checkbox"/>
12	<input type="checkbox"/>
13	<input type="checkbox"/>
14	<input checked="" type="checkbox"/>
15	<input type="checkbox"/>
16	<input checked="" type="checkbox"/>
17	<input checked="" type="checkbox"/>
18	<input type="checkbox"/>
19	<input checked="" type="checkbox"/>
20	<input checked="" type="checkbox"/>
21	<input type="checkbox"/>
22	<input type="checkbox"/>

Execution Time [min]:

Skill required:

0

1

2

Set of equipments:

<input type="checkbox"/> 1	<input type="checkbox"/> 6
<input checked="" type="checkbox"/> 2	<input type="checkbox"/> 7
<input checked="" type="checkbox"/> 3	<input type="checkbox"/> 8
<input type="checkbox"/> 4	<input type="checkbox"/> 9
<input type="checkbox"/> 5	<input type="checkbox"/> 10

Assembly Direction:

+x

+y

+z

-x

-y

-z

Go to

GenIAL

Figura 5.3: Interfaccia grafica delle operazioni elementari; esempio della 21° operazione del prodotto

- la/e precedenza/e tecnologiche;
- il tempo di esecuzione [min];
- la skill richiesta;
- il/gli equipment eventualmente utilizzati per svolgere l’operazione;
- la direzione di assemblaggio.

Si può inserire una sola skill e una sola direzione di assemblaggio per ciascuna operazione; per quanto riguarda gli equipment l’operazione può richiederne al massimo tre per la sua esecuzione.

Si inseriscono le operazioni a partire dalla prima e si digita il tasto “Save” per procedere da un’operazione alla successiva; con il tasto “Annul” si azzerano i valori inseriti circa l’operazione di riferimento. Se invece l’utente si rende conto di aver immesso dei dati sbagliati e vuole cancellarli può digitare il numero dell’operazione che vuole modificare nello spazio accanto a “Go to”, e poi con lo stesso ritornare successivamente a dove era arrivato.

Una volta inserite le caratteristiche dell’ultima operazione elementare, digitato il tasto “Save”, si apre una nuova interfaccia relativa ai parametri dell’algoritmo.

### **5.1.4 Interfaccia grafica dell’algoritmo**

A questa interfaccia si accede:

- dall’interfaccia grafica iniziale cliccando sul pulsante “Product already added” e indicando poi il prodotto precedentemente inserito cui l’utente fa riferimento;
- dalla precedente interfaccia una volta inserite tutte le operazioni elementari.

Come si vede in Figura 5.4 in questa interfaccia si immettono i parametri che descrivono l’algoritmo:

- il numero di iterazioni da eseguire;
- il numero di individui ovvero di cromosomi, tipicamente il doppio del numero di operazioni elementari;

- la frazione della popolazione, numero pari che rappresenta il numero di individui che saranno sottoposti a riproduzione;
- la probabilità associata all'operatore genetico di crossover;
- la probabilità associata all'operatore genetico di mutazione.

Parameter	Value
Number of iterations	100
Number of individuals	140
Population fraction	132
Probability of crossover	0.98
Probability of mutation	0.05

Save

GenIAL 1943

Figura 5.4: Interfaccia grafica dell'algoritmo

Come già spiegato nel Capitolo 3, essendo il crossover molto superiore in termini di risultati raggiunti rispetto alla mutazione, la probabilità del primo è generalmente molto più alta del secondo.

Per permettere all'utente di tenere nota riguardo tutti i dati e le informazioni inseriti, si è creata un'interfaccia di sintesi come ausilio anche per l'eventuale correzione di errori commessi durante il percorso.

### 5.1.5 Interfaccia grafica di sintesi dei dati

### Summary of the data added

#### Parameters of the product

Total number of operations

Production Rate [pieces/h]

Repositioning Time [min]

Line efficiency

#### Parameters of the algorithm

Number of iterations

Number of individuals

Population Fraction

Probability of crossover

Probability of mutation

#### Data operations

	Execution Time [min]	Skill required	Assembly Direction	
Enter Operation Number <input style="width: 50px;" type="text" value="21"/>	<input type="text" value="0.85"/>	<input type="text" value="1"/>	<input type="text" value="+y"/>	
	Equ 1	Equ 2	Equ 3	Precedences
	<input type="text" value="3"/>	<input type="text" value="2"/>	<input type="text" value="0"/>	<input style="width: 50px;" type="text" value="14"/> <input style="width: 50px;" type="text" value="16"/>

Figura 5.5: Interfaccia grafica di sintesi dei dati

Come si può osservare in Figura 5.5, in alto a sinistra si riportano i parametri del prodotto, in alto a destra quelli dell'algoritmo; nella finestra in basso è possibile digitare il numero dell'operazione elementare di cui si vuole avere informazioni e queste compariranno nei rispettivi riquadri di riferimento. Il riquadro delle precedenze, scorrendo con la freccia dello slider, permette di mostrare tutte le precedenze dell'operazione qualora ne avesse più di una.

A questo punto sono stati inseriti tutti i dati in input necessari alla programmazione, quindi l'utente può cliccare sul tasto "RUN" affinché tutti gli script che realizzano l'algoritmo genetico implementato siano eseguiti.

Nel dettaglio i risultati degli script riportano le seguenti informazioni:

- tempo necessario per assemblare il prodotto,  $T_p$ ;
- tempo ciclo,  $T_c$ ;
- $T_{max} = \max\{T_{si}\}$ ;
- limite inferiore per ciascun parametro obiettivo come descritto nel precedente capitolo:
  - $LB_N$ ;
  - $LB_S$ ;
  - $LB_E$ ;
  - $LB_D$ ;
  - $LB_{WV}$ ;
- limite superiore ( $UB$ ) per ciascun parametro obiettivo come descritto nel precedente capitolo:
  - $UB_N$ ;
  - $UB_S$ ;
  - $UB_E$ ;
  - $UB_D$ ;
  - $UB_{WV}$ ;
- sequenza ottimale, la miglior soluzione ottenuta durante le iterazioni, data dalla sequenza di operazioni elementari da eseguire per il montaggio del prodotto nel rispetto delle precedenze tecnologiche;
- numero  $N$  di stazioni di lavoro necessarie allo svolgimento delle operazioni elementari;
- distribuzione delle operazioni fra le  $N$  stazioni di lavoro;
- skill  $S$  necessaria per ogni stazione di lavoro;
- equipment  $E$  che devono essere presenti in ogni stazione di lavoro;
- cambio/i di direzione  $D$  che devono essere eseguiti in ogni stazione di lavoro e fra le stazioni;
- tempi di attesa  $T_{ai}$  presenti in ciascuna stazione

- tempo di attesa medio  $T_{am}$  ;
- Workload Variance  $WV$  presente fra le stazioni di lavoro;
- parametri obiettivo normalizzati:
  - $FN$ ;
  - $FS$ ;
  - $FE$ ;
  - $FD$ ;
  - $FWV$ ;
- fitness massima, ovvero il valore della funzione di fitness della sequenza ottimale;
- andamento durante le iterazioni di:
  - fitness massima ( $vf$ );
  - fitness media ( $vfm$ );
  - fitness minima ( $vfb$ );

Tutte le informazioni appena descritte permettono la realizzazione delle interfacce di output che si mostrano nei prossimi paragrafi.

### 5.1.6 Interfaccia grafica per la visualizzazione della miglior soluzione

Grazie all'interfaccia per la visualizzazione della miglior soluzione l'utente si può rendere conto graficamente dei risultati ottenuti dall'algoritmo.

In Figura 5.6 si mostra:

- l'istogramma con la distribuzione per ogni stazione di lavoro delle operazioni elementari nell'ordine della sequenza ottimale. Alla sinistra dell'istogramma si indica il valore del  $T_{max}$  ;
- l'andamento e il valore delle tre funzioni di fitness (minima, media e massima) durante le iterazioni;
- una finestra nella quale si riassumono i parametri obiettivo normalizzati ed il valore della fitness massima della miglior soluzione.

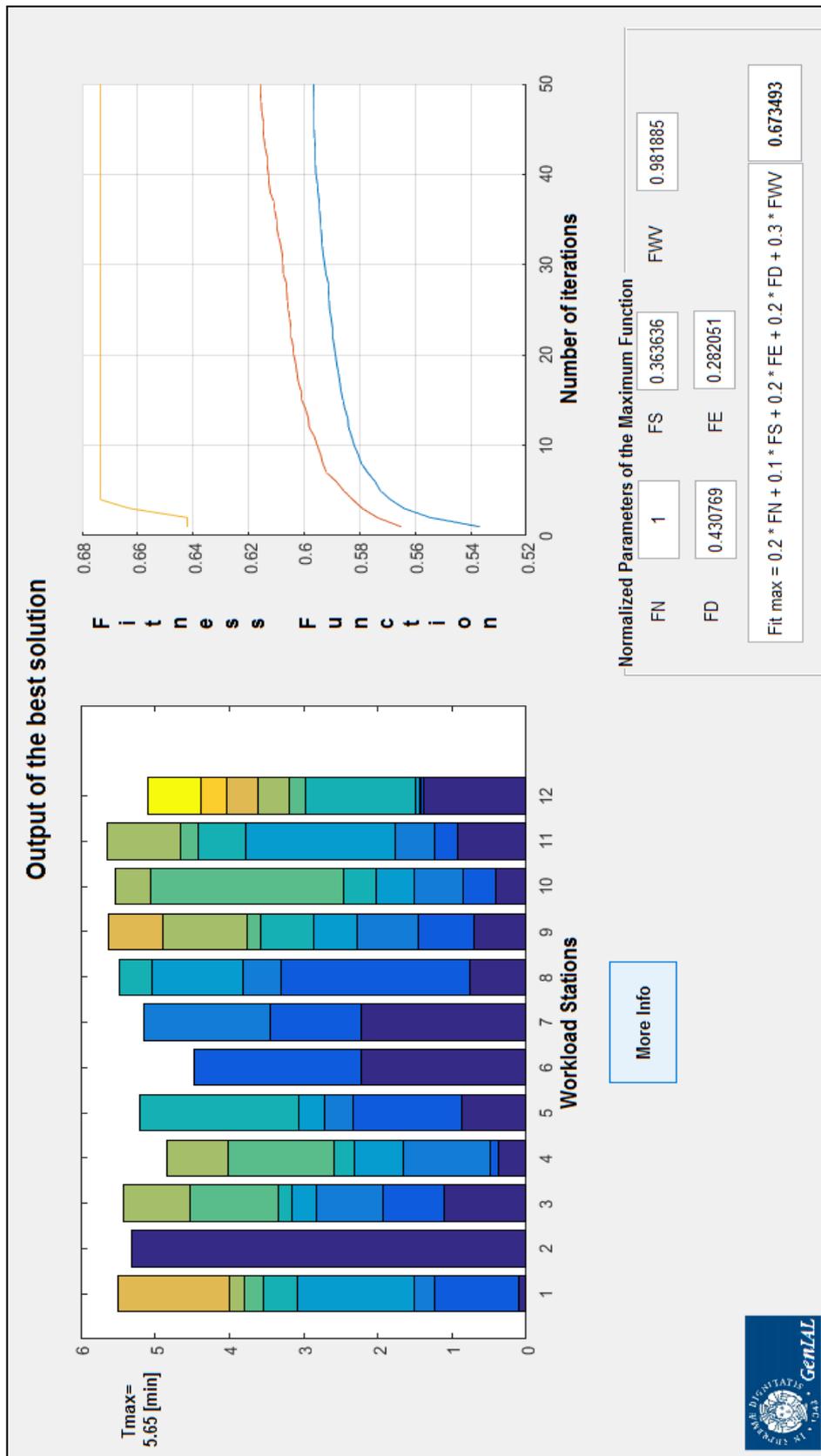


Figura 5.6: Interfaccia grafica per la visualizzazione della miglior soluzione

Cliccando sul pulsante “More Info” si accede ad un’ulteriore interfaccia che dettaglia le caratteristiche e le informazioni di ciascuna stazione di lavoro.

### 5.1.7 Interfaccia grafica per il dettaglio delle informazioni di ciascuna stazione di lavoro

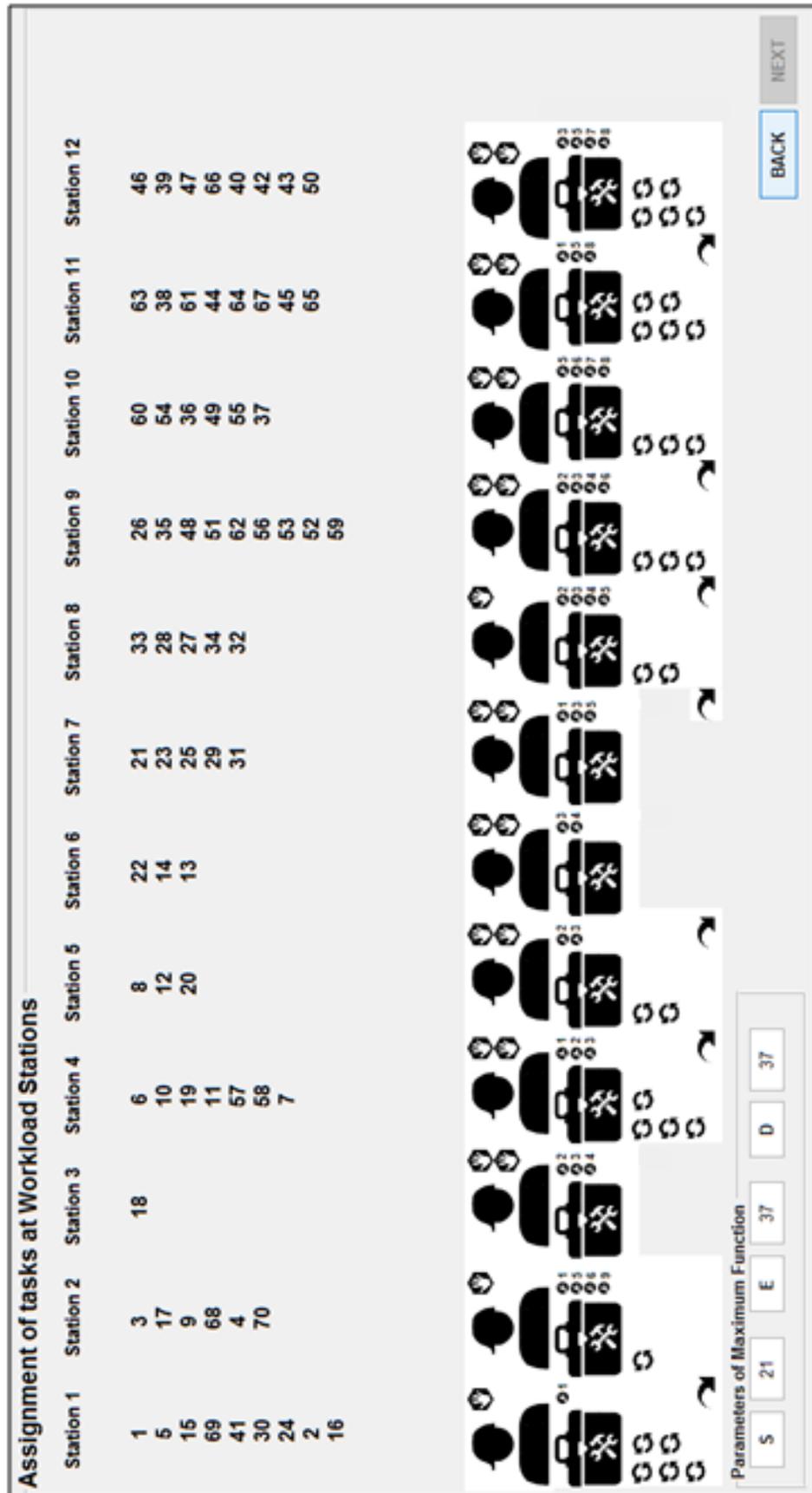


Figura 5.7: Interfaccia grafica per il dettaglio delle informazioni di ciascuna stazione di lavoro

Si mostra in Figura 5.7 l'interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione di lavoro. Dall'alto verso il basso si possono visualizzare per ciascuna stazione informazioni circa:

- le operazioni elementari che la costituiscono;
- la skill richiesta (icona con l'operatore, Figura 5.8, che può avere nessuna, una o due mani per indicare il livello di abilità necessario);

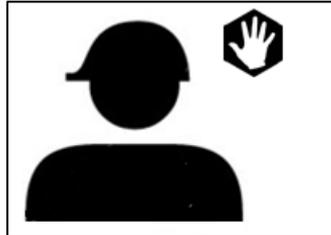


Figura 5.8: Esempio di icona che indica il livello 1 di abilità richiesto

- il/gli equipment che occorrono (icona con la cassetta degli attrezzi, Figura 5.9, che può avere da uno a dieci equipment, contrassegnati con il simbolo della chiave inglese);

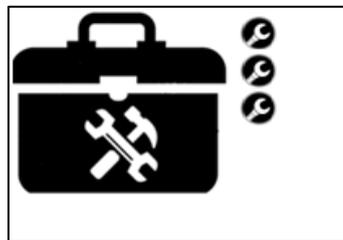


Figura 5.9: Esempio di icona che indica la necessità di 3 attrezzi nella stazione di lavoro

- il/i cambi di direzione di assemblaggio presenti (icona con le due frecce che si mostra tante volte quanti sono i cambi);

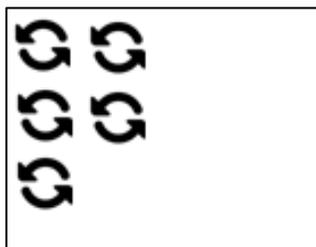


Figura 5.10: Esempio di icona che indica 5 cambi di direzione presenti nella stazione di lavoro

- per poter rappresentare l'eventuale cambio di direzione fra due stazioni si utilizza l'icona con la freccia verso destra posta fra le due stazioni di riferimento (Figura 5.11).

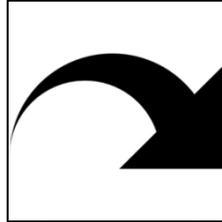


Figura 5.11: Icona che indica il cambio di direzione fra due stazioni di lavoro

In basso a sinistra dell'interfaccia si mostra una finestra in cui sono inseriti i valori dei parametri della funzione non normalizzati.

Se l'utente vuole tornare all'interfaccia precedente può cliccare su "BACK", in basso a destra. Il bottone "NEXT" è invece utilizzabile nel momento in cui la soluzione prevede più di 13 stazioni, e permette di accedere ad un'ulteriore interfaccia che dettaglia le stazioni successive (nell'esempio di Figura 5.7 non è possibile accedere al bottone in quanto la soluzione prevede 12 stazioni di lavoro).

## 5.2 Validazione

Si è deciso di andare a verificare la bontà dei limiti inferiori e superiori scelti e dei pesi attribuiti ai vari parametri della funzione obiettivo e lo si è fatto analizzando cinque casi di studio con l'algoritmo genetico implementato in Matlab. Per l'analisi si sono scelti cinque casi di studio che fossero differenti dal punto di vista del grafo di montaggio, ovvero più o meno vincolati. Un grafo si ritiene vincolato se le operazioni elementari sono legate fra loro da molte precedenze tecnologiche e quindi è bassa la possibilità di poterle scambiare fra le stazioni per cercare la soluzione migliore.

### 5.2.1 Validazione dei limiti inferiori e superiori

Al fine di verificare la validità dei  $LB$  e  $UB$  che sono stati scelti inizialmente (Capitolo 4) si è deciso di effettuare:

- uno studio della distribuzione dei valori assunti dai cinque obiettivi della funzione di fitness attraverso un certo numero di run dell'algoritmo genetico per ogni caso di studio;
- uno studio degli istogrammi di distribuzione globale, ovvero degli istogrammi che rappresentano l'andamento complessivo di ciascun parametro nei cinque casi di studio.

Si mostrano di seguito figure che riportano esempi della distribuzione dei valori assunti dagli obiettivi per due dei casi di studio analizzati, in particolare un caso di studio poco vincolato e uno più vincolato.

Al fine di chiarire la differenza fra i due casi di studio si riportano prima i grafi delle precedenze di ciascuno (Figura 5.12 e 5.13).

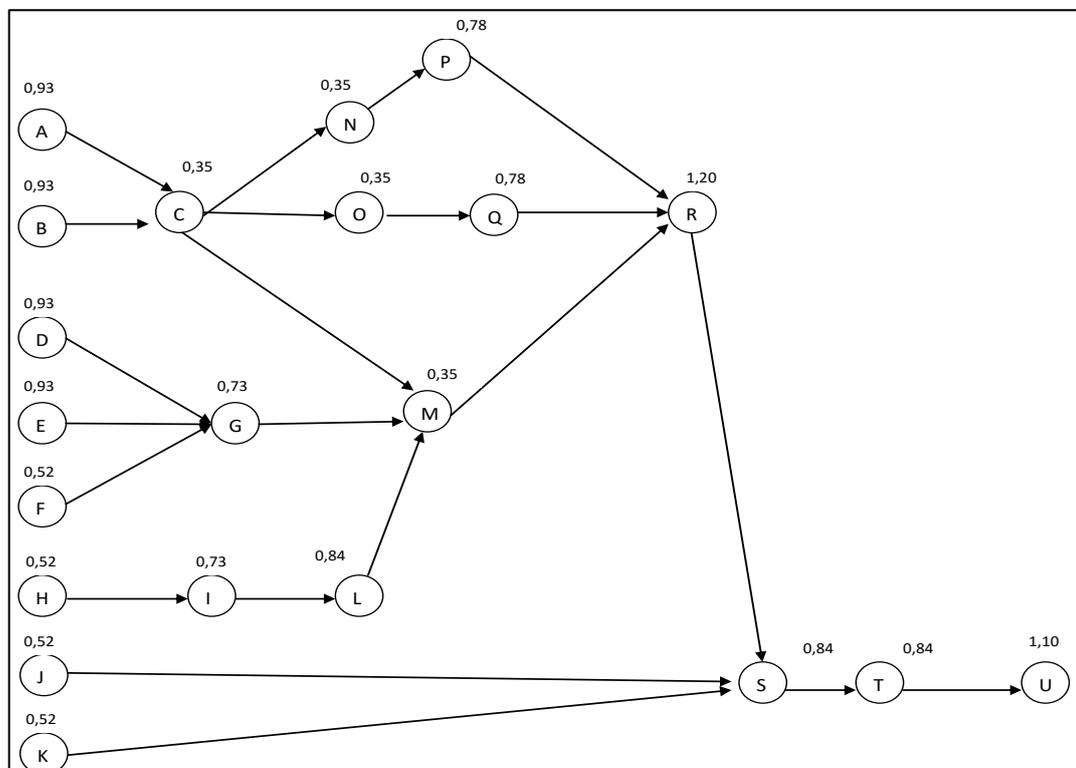


Figura 5.12: Grafo delle precedenze del caso di studio poco vincolato (caso A)

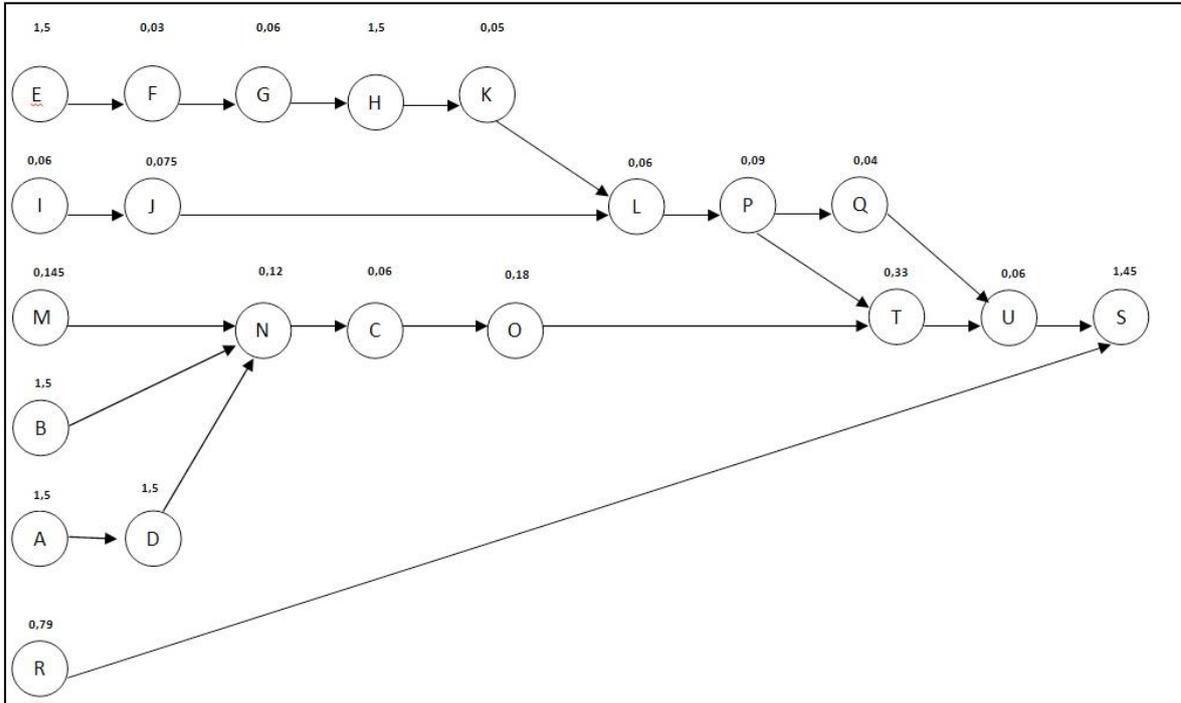


Figura 5.13: Grafo delle precedenze del caso di studio abbastanza vincolato (caso B)

La Figura 5.14 presenta la distribuzione dei parametri obiettivo per il caso A e la Figura 5.15 mostra nel dettaglio la distribuzione del parametro D.

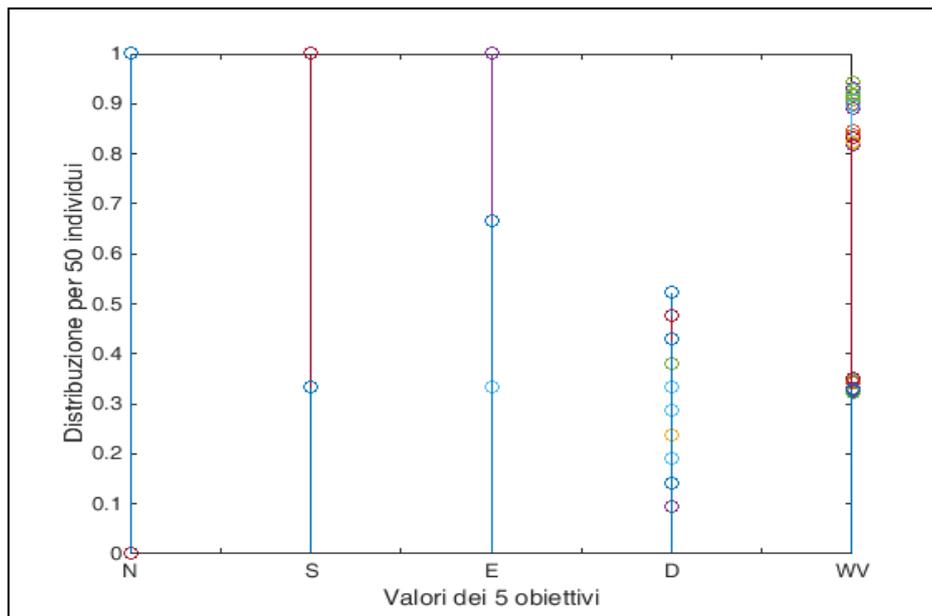


Figura 5.14: Distribuzione dei 5 parametri per 50 individui per il caso A

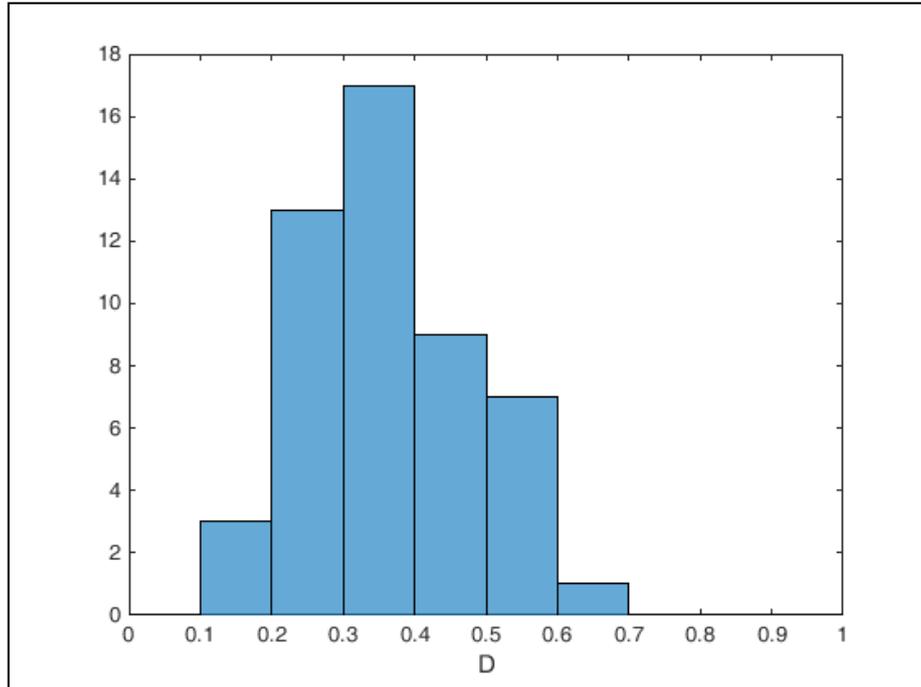


Figura 5.15: Distribuzione del parametro D per il caso A

La Figura 5.16 presenta invece la distribuzione dei parametri obiettivo per il caso B e la Figura 5.17 mostra nel dettaglio la distribuzione del parametro D.

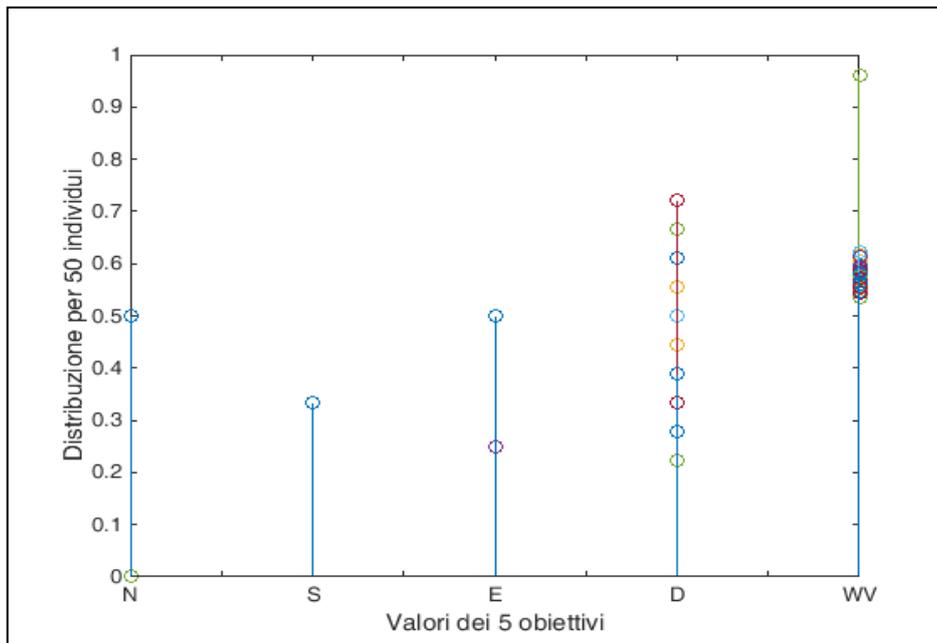


Figura 5.16: Distribuzione dei 5 parametri per 50 individui per il caso di studio B

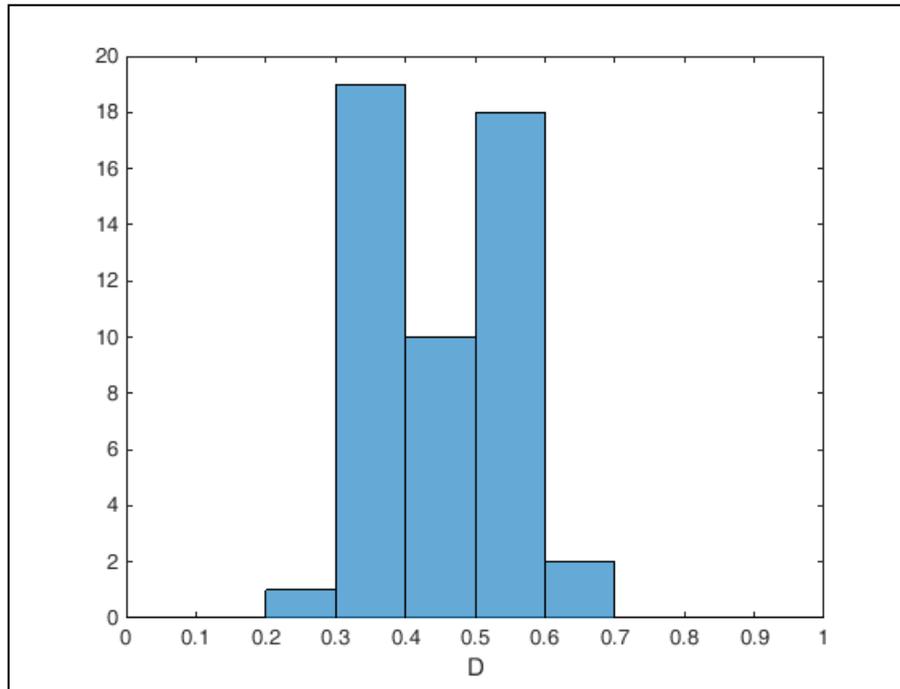


Figura 5.17: Distribuzione del parametro D per il caso B

In generale si può affermare che il risultato dell'analisi è una buona distribuzione dei valori che confermano la bontà dei  $LB$  e  $UB$  scelti, ma con differenza fra la tipologia del grafo di partenza. Come si nota nelle Figure 5.15 e 5.17, la prima presenta una distribuzione che si avvicina di più ad una gaussiana rispetto alla seconda.

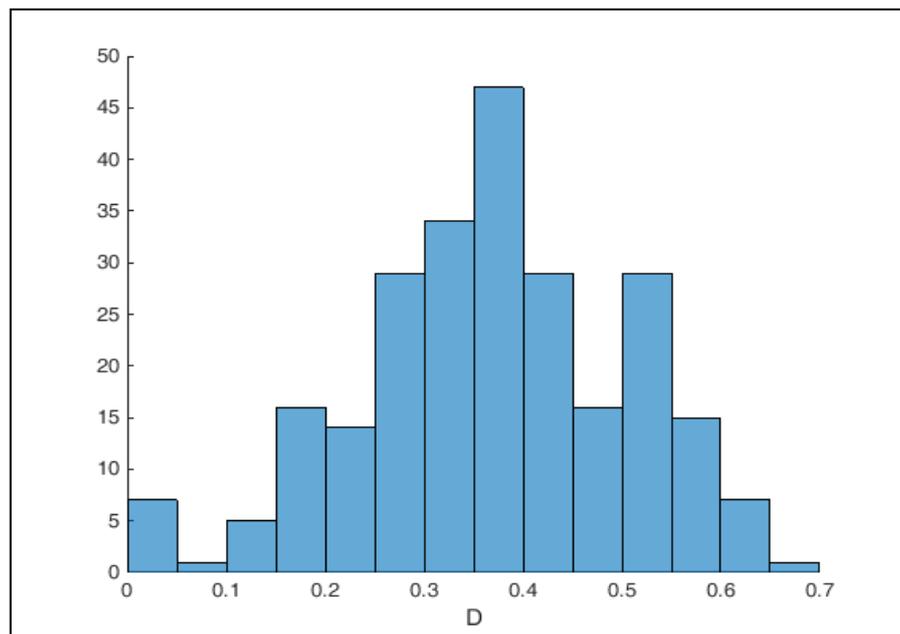


Figura 5.18: Distribuzione globale del parametro D

Per quanto riguarda gli istogrammi di distribuzione globale, anche in questo caso si è dimostrato un buon andamento dei parametri; si riporta come esempio una distribuzione globale del parametro D in Figura 5.18.

### 5.2.2 Validazione dei pesi da attribuire alla fitness

Con lo scopo di individuare la miglior distribuzione dei pesi dei vari parametri da attribuire alla fitness, si è concentrato lo studio sul caso di studio A e si sono combinati diversi valori dei pesi per poter analizzare più configurazioni.

Combinazioni	Pesi dei vari parametri obiettivo					Funzione Fitness
	N	S	E	D	WV	
1	1	0	0	0	0	<b>0.67</b>
2	0	1	0	0	0	<b>0.55</b>
3	0	0	1	0	0	<b>0.8</b>
4	0	0	0	1	0	<b>0.75</b>
5	0	0	0	0	1	<b>0.98</b>
6	0.5	0.5	0	0	0	<b>0.56</b>
7	0.5	0	0.5	0	0	<b>0.73</b>
8	0.5	0	0	0.5	0	<b>0.78</b>
9	0.5	0	0	0	0.5	<b>0.83</b>
10	0	0.5	0.5	0	0	<b>0.58</b>
11	0	0.5	0	0.5	0	<b>0.62</b>
12	0	0.5	0	0	0.5	<b>0.68</b>
13	0	0	0.5	0.5	0	<b>0.75</b>
14	0	0	0.5	0	0.5	<b>0.84</b>
15	0	0	0	0.5	0.5	<b>0.82</b>
16	0.2	0.2	0.2	0.2	0.2	<b>0.66</b>
17	0.4	0.2	0.1	0.2	0.1	<b>0.64</b>
18	0.4	0.1	0.2	0.1	0.2	<b>0.7</b>
19	0.2	0.1	0.2	0.2	0.3	<b>0.72</b>

Tabella 5.1: Combinazioni utilizzate per la scelta della migliore configurazione dei pesi da attribuire alla fitness

In Tabella 5.1 si mostrano tutte le configurazioni dei pesi utilizzate nello studio del caso A.

La scelta finale della combinazione dei pesi non è stata dettata solamente dal valore più alto raggiunto dalla funzione di fitness, ma anche da un giudizio personale riguardo l'importanza dei singoli obiettivi.

La combinazione che si predilige per il presente lavoro di tesi è l'ultima in quanto valorizza il parametro  $WV$  che risulta essere quello che ha permesso i valori maggiori di fitness (combinazione 5,9,14,15 della Tabella 5.1), ma allo stesso tempo prevede un buon equilibrio con e fra gli altri parametri obiettivo. Non solo, tale combinazione rispecchia la scaletta di importanza che si attribuisce ai parametri da un punto di vista logico del bilanciamento di una linea di montaggio.

## Capitolo 6 Applicazione: Casi di studio

Nel presente ed ultimo capitolo si mostrano i risultati ottenuti dall'applicazione di "GenIAL" a due casi di studio. Si sono scelti due casi di studio che potessero evidenziare al meglio le potenzialità dell'algoritmo genetico implementato:

- il primo caso di studio (Carburetor Keihin 19 PD, paragrafo 6.1) rappresenta un esempio di prodotto tratto dalla tipica realtà industriale ed inoltre, come si può osservare dal grafo delle precedenze (Figura 6.3), si tratta di un caso poco vincolato, che consente perciò un'elevata flessibilità nell'assegnazione delle operazioni elementari alle stazioni di lavoro;
- il secondo caso di studio (Tonge, paragrafo 6.2) proviene dalla letteratura e quindi si è scelto affinché potessimo paragonare l'approccio genetico sviluppato ad alcuni dei metodi che sono stati proposti negli anni per la risoluzione dei problemi di bilanciamento delle linee di assemblaggio.

### 6.1 Caso di studio: Carburetor Keihin 19 PD

#### 6.1.1 Descrizione

Il primo caso di studio che si analizza è un carburatore (Figura 6.1), Carburetor Keihin 19 PD, costituito da 27 componenti.

Si mostra in Figura 6.2 il modello esploso del carburatore e in Tabella 6.1 le 25 operazioni elementari necessarie per il completo assemblaggio del prodotto, il loro tempo di esecuzione e le precedenze tecnologiche.

La linea di assemblaggio del carburatore deve essere dimensionata e bilanciata secondo una domanda annua di 50.000 pezzi; inoltre la linea lavora per 50 settimane all'anno, 5 turni a settimana e 8 ore per turno. Si ipotizza un unico operatore per stazione, una efficienza di linea del 95 % e un tempo di riposizionamento di 0.05 min.

Il rateo produttivo è perciò di:

$$R_p = 50.000 / (50 * 5 * 8) = 25 \text{ pezzi/ora.}$$

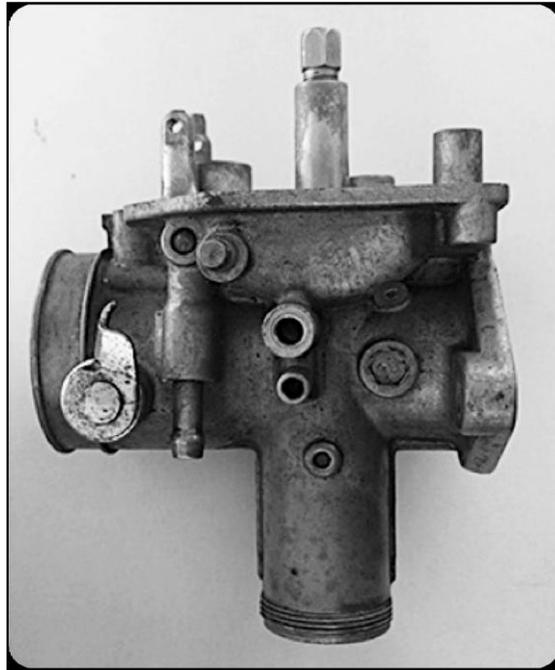


Figura 6.1: Carburatore Keihin 19 PD

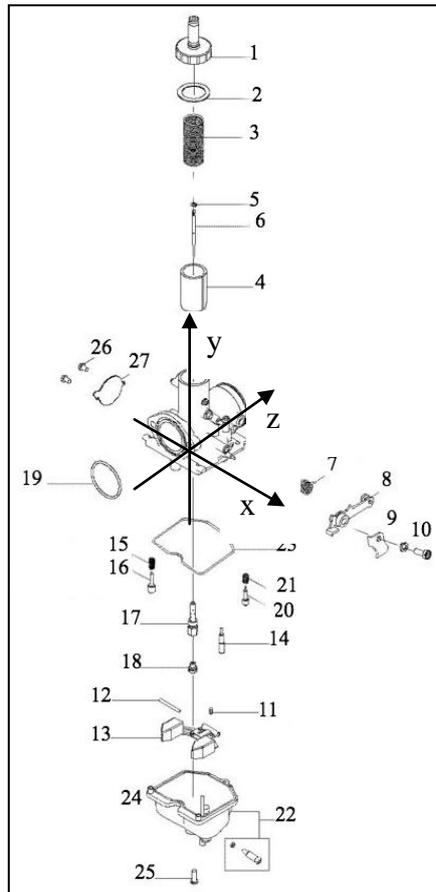


Figura 6.2: Modello esploso Carburatore Keihin 19 PD

<b>N</b>	<b>Tasks</b>	<b><math>T_{ek}</math> [min]</b>	<b>Precedences</b>
1	Clip (n°5) on Jet Needle (n°6)	0,16	-
2	Insert n° 5-6 in Throttle Valve n°4	0,1	1
3	Insert n° 4-5-6 in Body n°0	0,14	2
4	Insert Throttle Spring n°3 in n° 0-4-5-6	0,1	3
5	Insert Gasket n° 2 in Tops n°1	0,1	-
6	Scew down n° 1-2 on Body n°0	0,34	4,5
7	Cap Cable Holder n°7 on Body n°0	0,14	-
8	Cable Adjuster n°8 on n° 0-7	0,14	7
9	Nut/Cable Adjuster n°9 on n° 0-7-8	0,16	8
10	Screw/F.Bowl and Cap Cable Holder n°10 on n° 0-7-8-9	0,34	9
11	Gasket n°19 on Body n°0	0,1	-
12	2 Screws/F.Bowl n°26 on Throttle Valve n°27	0,67	-
13	Screw down Needle Jet n°17 on Body n°0	0,34	-
4	Screw down Main Jet n°18 on n° 0-17	0,34	13
15	Spring n°15 on Air Adjuster Screw n°14	0,1	-
16	Screw down n° 14-15 on Body n°0	0,34	15
17	Screw down Slow Jet n°16 on Body n°0	0,34	-
18	Float Valve n°11 on Body n°0	0,1	-
19	Float n°13 on Body n°0 with Float Pin n°12	0,42	18
20	Stick Gasket/Bowl n°23 on Float Bowl n°24	0,1	-
21	3 Screws/F.Bowl Cap Cable n°25 on n° 23-24 and Body n°0	0,85	14,16,17,19,20
22	Screw down Plug Screw Washer n°22 on Float Bowl n°24	0,34	-
23	Spring n°21 on Throttle Stop Screw n°20	0,14	-
24	Screw down n° 20-21 on Body n°0	0,34	23
25	Final Test	1,5	6,10,11,12,21,22,24

Tabella 6.1: Operazioni elementari per l'assemblaggio del Carburatore Keihin 19 PD

Si mostra inoltre in Figura 6.3 il grafo delle precedenze del prodotto.

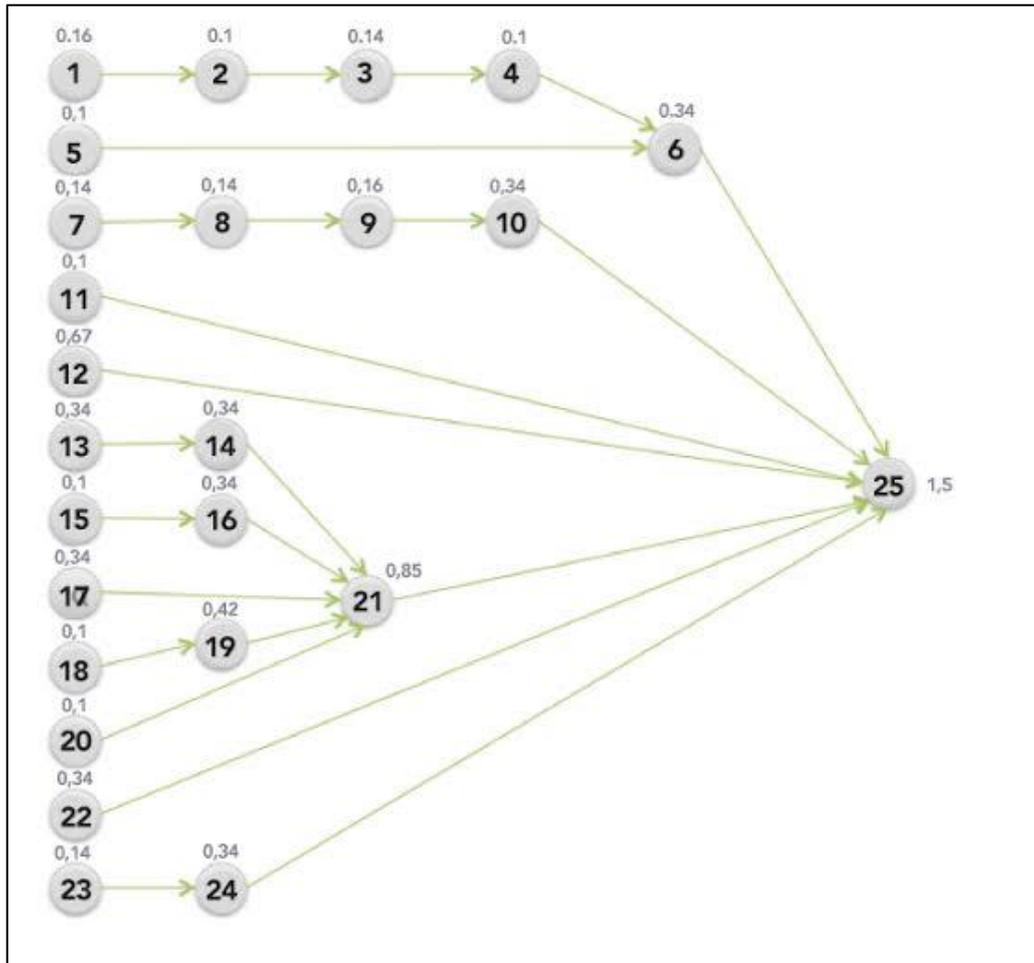


Figura 6.3: Grafo delle precedenze del Carburatore Keihin 19 PD

### 6.1.2 Implementazione in “GenIAL”

Nel Capitolo 5 si sono descritti i passi necessari affinché “GenIAL” possa eseguire l’algoritmo genetico.

La prima fase prevede l’immissione dei parametri del prodotto, già indicati nel precedente paragrafo. Dopodiché, la prima volta che si studia il prodotto, si inseriscono le informazioni circa ciascuna operazione elementare. Si mostra in Tabella 6.2 la matrice che rappresenta la prima configurazione scelta per il carburatore.

La prima configurazione considera i seguenti criteri nella scelta dei parametri  $D$ ,  $E$  e  $S$ :

- la direzione di assemblaggio si basa su di un sistema di riferimento  $xyz$  con origine nel centro del carburatore (disegnato sulla Figura 6.2);

- gli equipment inseriti sono quelli utilizzati tipicamente nella produzione industriale di un carburatore: Eq 1= Avvitatore, Eq 2 = Pinza, Eq 3 = Chiave inglese ecc;
- la skill si assegna sulla base di una valutazione di difficoltà di ogni specifica operazione, considerando quindi anche il tempo di esecuzione della stessa e il/ gli equipment richiesti per eseguirla.

Operazione	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Direzione
1	0,16	1	1	-	-	-x
2	0,1	0	1	-	-	-y
3	0,14	0	-	-	-	-y
4	0,1	0	-	-	-	-y
5	0,1	0	-	-	-	-y
6	0,34	0	-	-	-	-y
7	0,14	0	1	-	-	-x
8	0,14	0	-	-	-	-x
9	0,16	1	1	-	-	-x
10	0,34	1	2	-	-	-x
11	0,1	0	-	-	-	-z
12	0,67	1	2	-	-	+x
13	0,34	1	4	-	-	+y
14	0,34	1	4	-	-	+y
15	0,1	0	-	-	-	-y
16	0,34	1	4	-	-	+y
17	0,34	1	3	-	-	+y
18	0,1	0	-	-	-	+y
19	0,42	1	-	-	-	+y
20	0,1	0	-	-	-	-y
21	0,85	2	2	-	-	+y
22	0,34	1	3	-	-	+z
23	0,14	0	-	-	-	+y
24	0,34	1	2	-	-	+y
25	1,5	2	5	-	-	-y

Tabella 6.2: Prima configurazione del Carburatore Keihin 19 PD

L'ultima fase che precede l'esecuzione dell'algoritmo genetico prevede l'inserimento dei parametri dello stesso. Nel caso in esame:

- numero di iterazioni: 50, numero di partenza che si immagina possa essere adatto al numero di operazioni del caso;
- numero di individui: 50, si sceglie inizialmente come il doppio del numero di operazioni elementari;

- frazione della popolazione: 48;
- probabilità di crossover: 0.98; come già spiegato precedentemente, l'operatore genetico di crossover genera soluzioni molto superiori in termini di risultati rispetto alla mutazione, perciò si utilizza una probabilità molto più alta;
- probabilità di mutazione: 0.05.

A questo punto “GenIAL” è in grado di svolgere l'algoritmo e di restituire i risultati ottenuti. “GenIAL” può eseguire con un “run” tante generazioni quante ne desidera l'utente, compararle, e restituire la miglior soluzione fra quelle valutate. Nel caso in esame si imposta il programma in modo tale da produrre automaticamente 10 generazioni per ciascun “run”.

Nel presente lavoro di tesi si sono svolte prove su differenti configurazioni, create variando i parametri  $S$  ed  $E$  a partire da quella iniziale; si descrivono di seguito i criteri per i quali si sono scelte le successive configurazioni:

- a partire dalla configurazione base di  $S$  ed  $E$ , si varia per primo  $S$  lasciando inalterati gli altri parametri. Si varia  $S$  per tre volte, e lo si fa minimizzando le skill nel primo caso, variandole mediamente nel secondo, e massimizzandole nel terzo caso;
- stesso procedimento per gli equipment  $E$ , sempre a partire dalla configurazione di base;
- ad ognuna delle tre configurazioni create con le variazioni di  $S$ , si associano le tre configurazioni create con le variazioni di  $E$ .

Seguendo questi passi si ottengono 16 configurazioni da poter analizzare con “GenIAL”.

Durante lo studio, si è poi evidenziata una tendenza della funzione di fitness ad appiattirsi prima di raggiungere il numero di iterazioni richiesto, quindi si è deciso di ripetere le prove delle 16 configurazioni diminuendo il numero di iterazioni e variando leggermente le probabilità degli operatori genetici.

Il numero totale dei casi studiati è quindi di 32. Si riportano nel prossimo paragrafo i risultati di alcune configurazioni, quelli considerati più rilevanti; si allegano invece i risultati della maggior parte delle configurazioni in Appendice A.

### 6.1.3 Risultati

Il primo caso che si mette in evidenza in questo paragrafo è la miglior soluzione ottenuta per la configurazione che massimizza le skill e minimizza gli equipment rispetto alla configurazione di partenza. Si riporta in Tabella 6.3 la configurazione, in Figura 6.4 l'interfaccia grafica per la visualizzazione della miglior soluzione ed in Figura 6.5 quella con il dettaglio delle caratteristiche di ciascuna stazione di lavoro.

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,16	1	0	0	0	-x
2	0,1	2	1	0	0	-y
3	0,14	1	0	0	0	-y
4	0,1	0	0	0	0	-y
5	0,1	1	0	0	0	-y
6	0,34	1	0	0	0	-y
7	0,14	0	0	0	0	-x
8	0,14	0	0	0	0	-x
9	0,16	1	0	0	0	-x
10	0,34	2	1	0	0	-x
11	0,1	0	0	0	0	-z
12	0,67	1	1	0	0	+x
13	0,34	1	0	0	0	+y
14	0,34	2	0	0	0	+y
15	0,1	0	0	0	0	-y
16	0,34	2	0	0	0	+y
17	0,34	2	2	0	0	+y
18	0,1	0	0	0	0	+y
19	0,42	1	0	0	0	+y
20	0,1	1	0	0	0	-y
21	0,85	2	1	0	0	+y
22	0,34	1	2	0	0	+z
23	0,14	1	0	0	0	+y
24	0,34	1	2	0	0	+y
25	1,5	2	3	0	0	-y

Tabella 6.2: Configurazione che massimizza le skill e massimizza gli equipment rispetto a quella di partenza

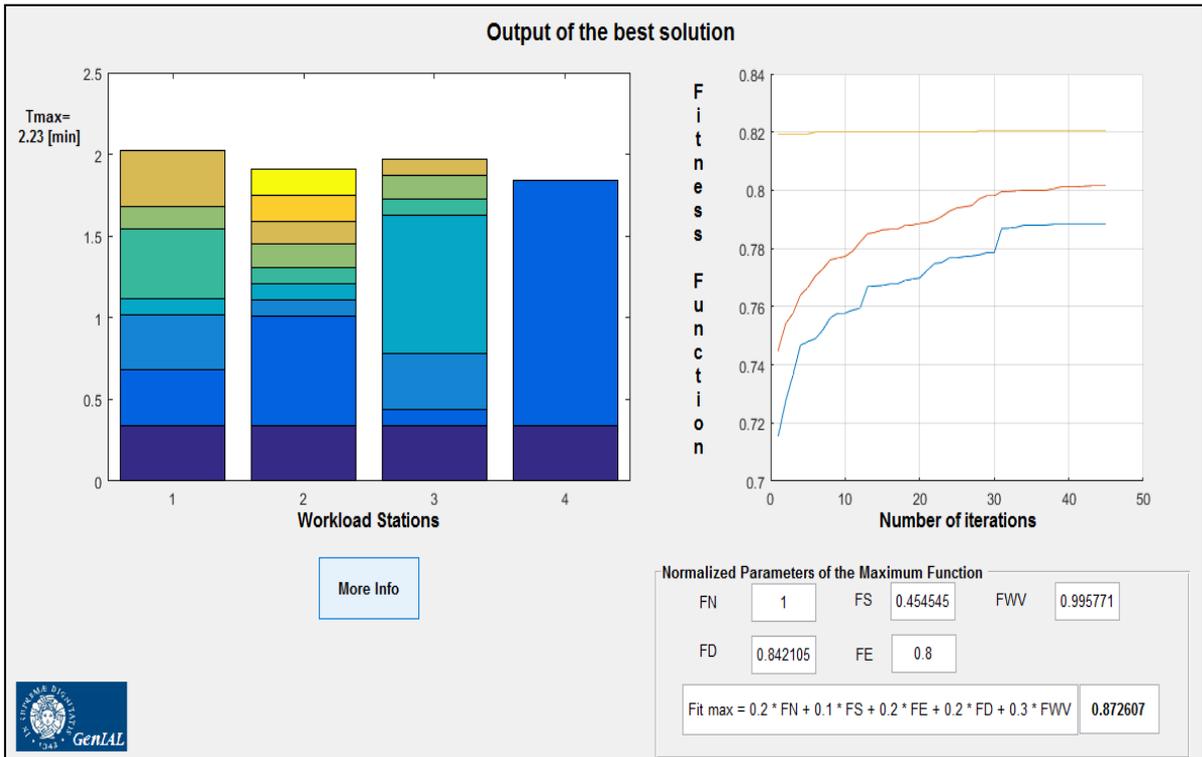


Figura 6.4: Miglior soluzione per la configurazione che massimizza le skill e minimizza gli equipment rispetto alla configurazione di partenza

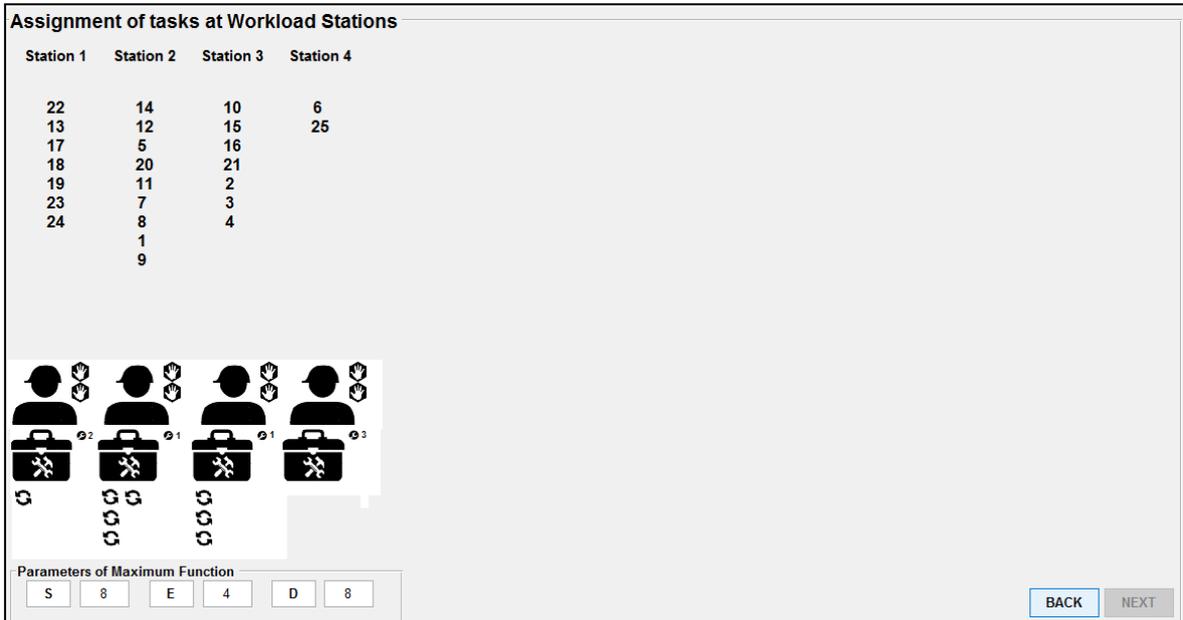


Figura 6.5: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione di lavoro per la configurazione che massimizza le skill e minimizza gli equipment rispetto alla configurazione di partenza

Per sottolineare la bontà della soluzione individuata da “GenIAL” rispetto a quella che si può ottenere con un metodo euristico, si risolve la medesima configurazione con il metodo LCR.

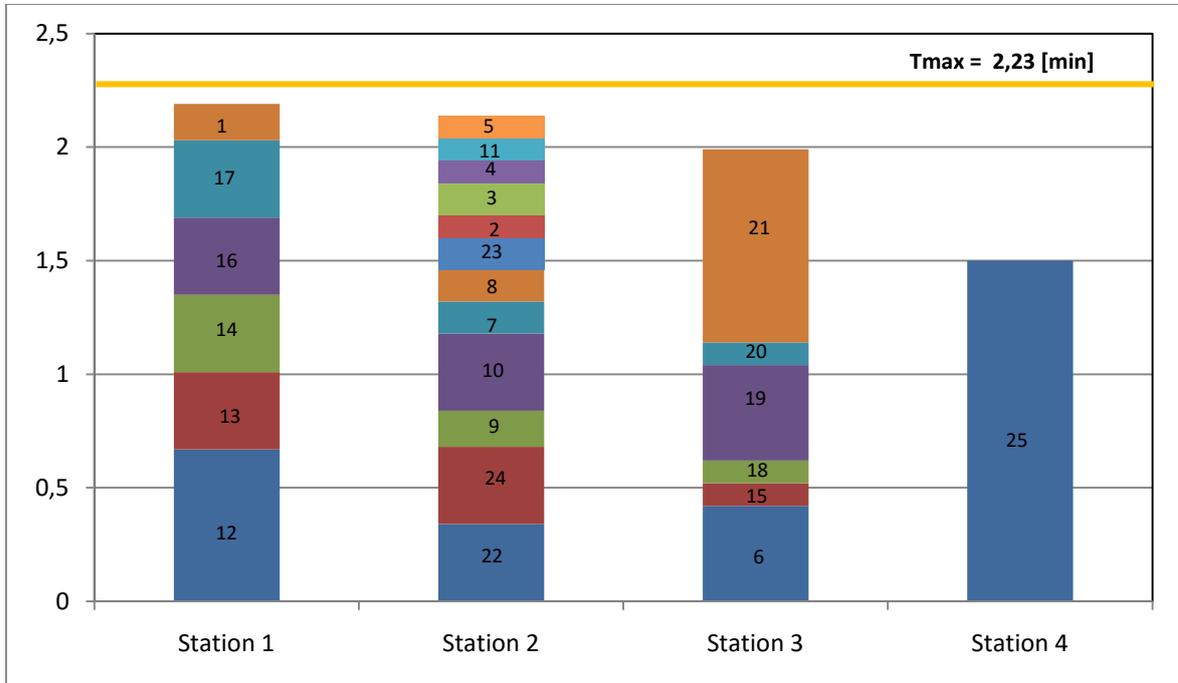


Figura 6.6: Istogramma della soluzione con metodo LCR

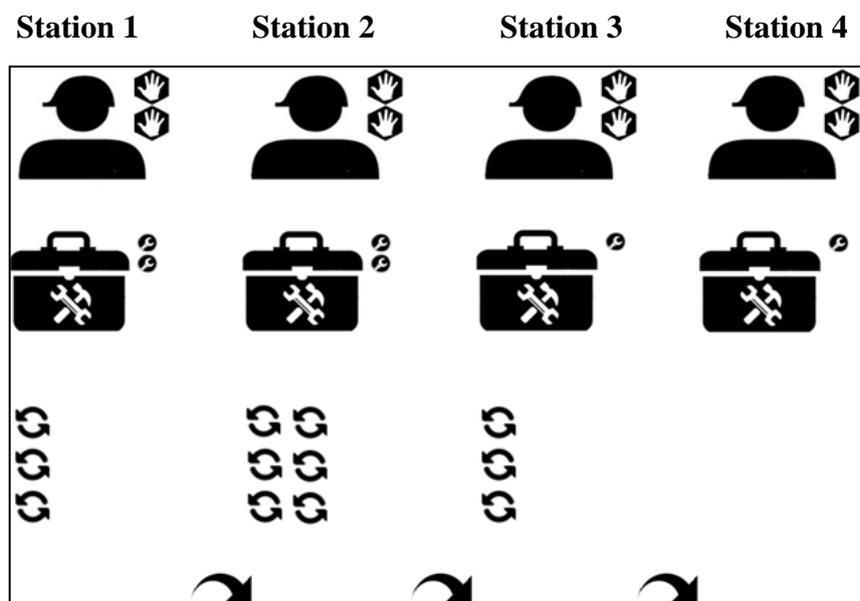


Figura 6.7: Caratteristiche di ciascuna stazione di lavoro ottenute con il metodo LCR per la configurazione che massimizza le skill e minimizza gli equipment rispetto alla configurazione di partenza

Come si può notare soprattutto dalle differenze fra le Figure 6.5 e 6.7, a parte le skill che rimangono invariate per entrambi i metodi, gli equipment e i cambi di direzione sono superiori se si utilizza il metodo LCR; nel dettaglio:

- con “GenIAL” la soluzione richiede un equipment per stazione, mentre con LCR nelle prime due stazioni sono necessari due equipment;
- con “GenIAL” i cambi di direzione sono inferiori nelle prime due stazioni, e non sono necessari cambi di direzione fra le stazioni come invece si nota nella soluzione con metodo LCR.

Si calcola infine la funzione di fitness che si ottiene con la soluzione del metodo LCR:

- si determinano i limiti inferiori e superiori di ciascun parametro;
- si calcolano i parametri normalizzati utilizzando la formula già descritta nel Capitolo 4:

$$F_i = 1 + (LB - x)/(UB - LB)$$

- $FN = 1$ ;
- $FS = 0.45$ ;
- $FD = 0.42$ ;
- $FE = 0.4$ ;
- $FWV = 0.93$ ;
- la funzione di fitness si calcola come:

$$F = 0.2 * FN + 0.1 * FS + 0.2 * FD + 0.2 * FE + 0.3 * FWV = 0.688$$

I valori dei parametri normalizzati confermano quanto affermato sulle differenze fra i due metodi utilizzati per risolvere la stessa configurazione: solo il valore del parametro  $FS$  rimane invariato, gli altri sono inferiori e perciò anche la funzione di fitness (0.688 invece di 0.872).

Il secondo caso che si descrive in questo paragrafo è la miglior soluzione ottenuta per la configurazione che minimizza le skill rispetto alla configurazione di partenza. Si riporta anche per questa soluzione in Tabella 6.3 la configurazione, in Figura 6.8 l’interfaccia grafica per la visualizzazione della miglior soluzione, ed in Figura 6.9 quella con il dettaglio delle caratteristiche di ciascuna stazione di lavoro.

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,16	0	1	0	0	-1
2	0,1	0	1	0	0	-2
3	0,14	0	0	0	0	-2
4	0,1	0	0	0	0	-2
5	0,1	0	0	0	0	-2
6	0,34	0	0	0	0	-2
7	0,14	0	1	0	0	-1
8	0,14	0	0	0	0	-1
9	0,16	0	1	0	0	-1
10	0,34	1	2	0	0	-1
11	0,1	0	0	0	0	-3
12	0,67	0	2	0	0	1
13	0,34	0	0	0	0	2
14	0,34	0	4	0	0	2
15	0,1	0	0	0	0	-2
16	0,34	0	4	0	0	2
17	0,34	0	3	0	0	2
18	0,1	0	0	0	0	2
19	0,42	0	0	0	0	2
20	0,1	0	0	0	0	-2
21	0,1	1	2	0	0	2
22	0,34	0	3	0	0	3
23	0,14	0	0	0	0	2
24	0,34	0	2	0	0	2
25	1,5	2	5	0	0	-2

Tabella 6.3: Configurazione che minimizza le skill rispetto a quella di partenza

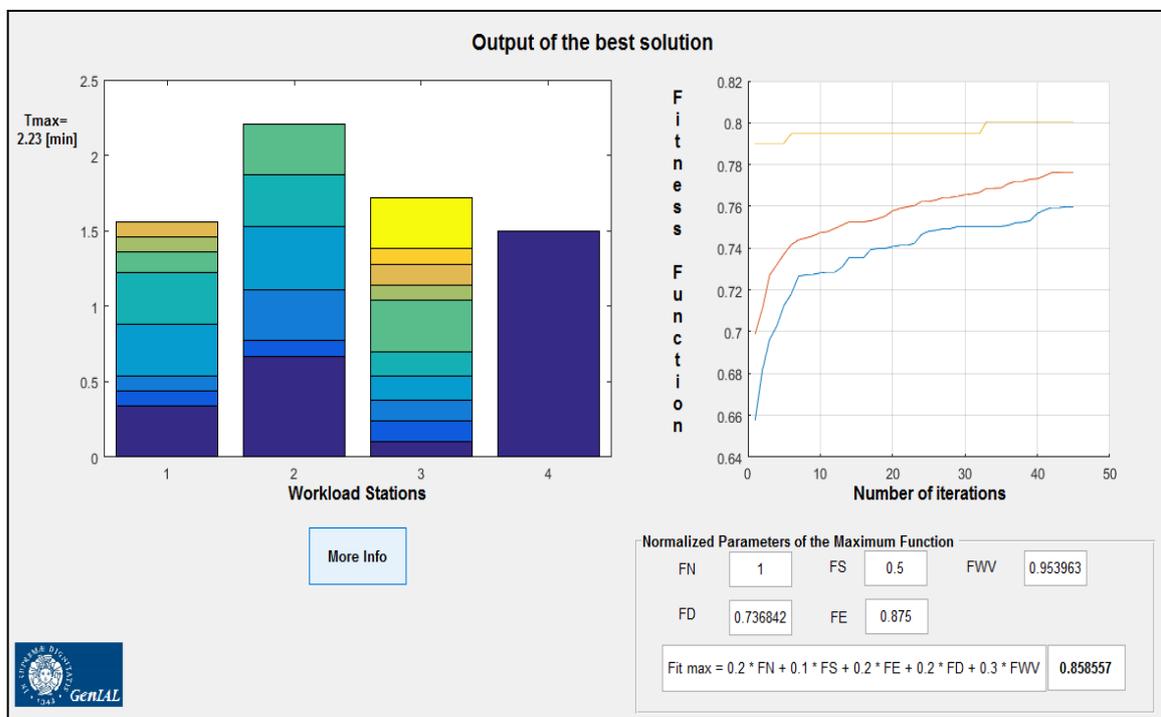


Figura 6.8: Miglior soluzione per la configurazione che minimizza le skill rispetto alla configurazione di partenza

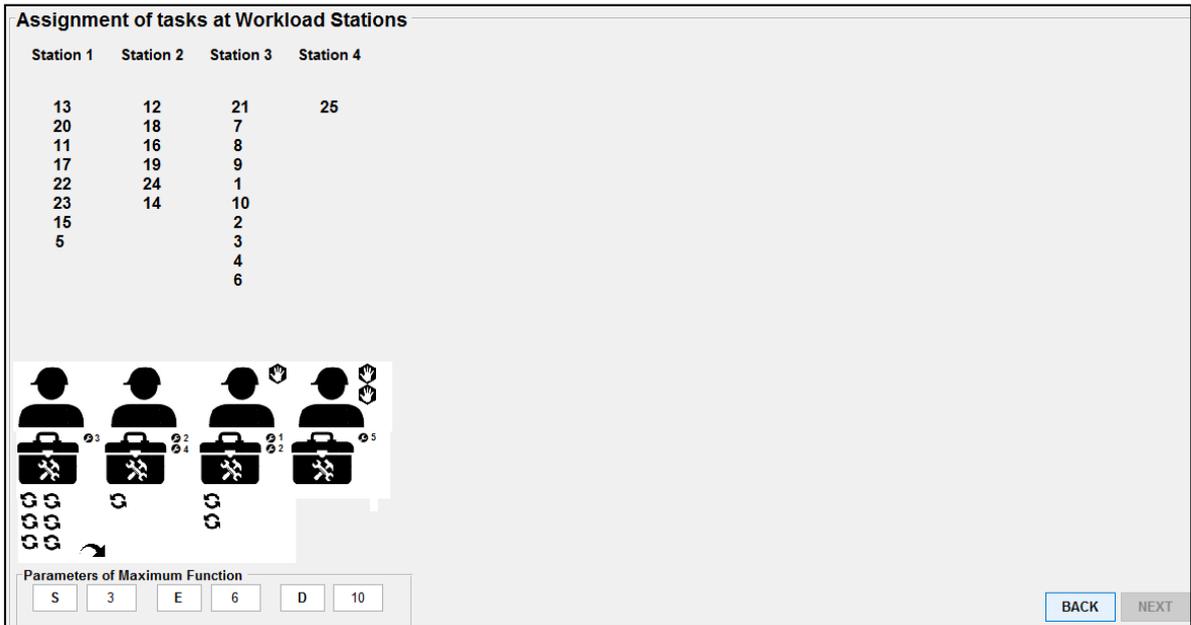


Figura 6.9: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione di lavoro per la configurazione che minimizza le skill rispetto alla configurazione di partenza

Risolvendo la configurazione con il metodo LCR si ottiene, anche in questo caso, una soluzione peggiore rispetto a quella ottenuta con “GenIAL”. In Figura 6.10 si mostrano le caratteristiche di ciascuna stazione di lavoro.

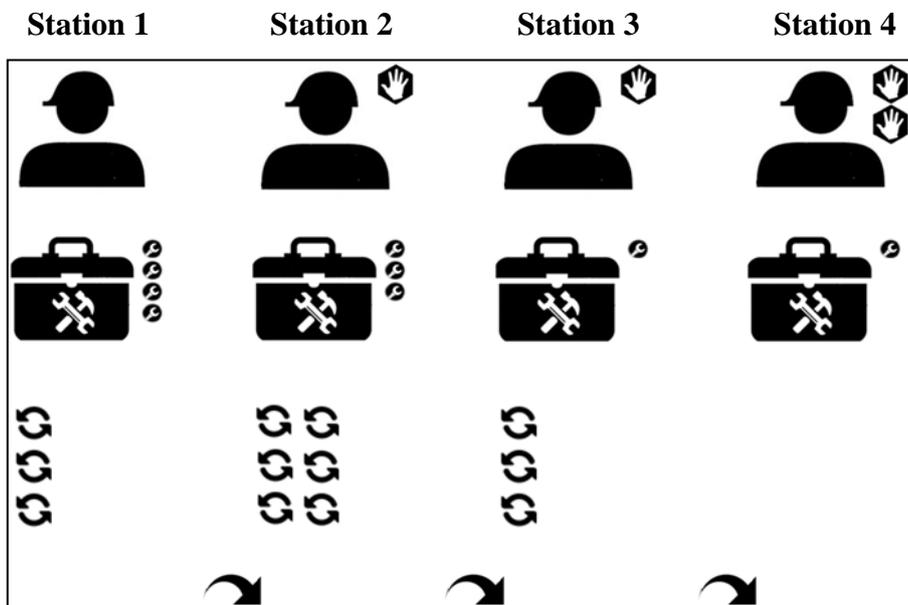


Figura 6.10: Caratteristiche di ciascuna stazione di lavoro ottenute con il metodo LCR per la configurazione che minimizza le skill rispetto alla configurazione di partenza

Le differenze fra Figura 6.11 e 6.12 evidenziano:

- una skill in più necessaria nella seconda stazione di lavoro;
- tre equipment in più nella prima stazione, uno nella seconda, uno in meno nella terza e la stessa nella quarta: in totale tre equipment in più;
- quindici cambi di direzione totali contro i dieci della soluzione con “GenIAL”.

Si calcola la funzione di fitness che si ottiene con la soluzione del metodo LCR ed il valore risulta essere di 0.67 invece di 0.86 dell’algoritmo genetico implementato.

Ciascuno dei 32 casi analizzati nel presente lavoro è stato risolto anche con il metodo LCR; si presenta in Tabella 6.4 una sintesi dei valori dei parametri normalizzati e delle fitness ottenuti per entrambi i metodi utilizzati. Quello che emerge è che, per ogni configurazione, i risultati ottenuti con l’euristica sono inferiori e quindi peggiori di quelli ottenuti con l’algoritmo implementato.

Conf	GenIA L FN	LC R FN	GenIA L FS	LC R FS	GenIA L FE	LC R FE	GenIA L FD	LC R FD	GenIA L FWV	LCR FW V	GenIA L Fitness	LCR Fitness
1	1	1	0,33	0,33	1	0,56	0,53	0,42	0,95	0,93	0,8243	0,708
2	1	1	0,5	0	0,875	0,5	0,53	0,42	0,97	0,94	0,8221	0,67
3	1	1	0,44	0,44	0,875	0,63	0,63	0,42	0,95	0,93	0,832	0,734
4	1	1	0,45	0,45	0,78	0,56	0,74	0,42	0,99	0,93	0,8475	0,722
5	1	1	0,33	0,33	1	0,25	0,58	0,42	0,99	0,93	0,8474	0,646
6	1	1	0,33	0,33	0,64	0,64	0,84	0,42	0,99	0,93	0,8293	0,724
7	1	1	0,33	0,33	0,75	0,58	0,68	0,42	0,98	0,93	0,8153	0,712
8	1	1	0,5	0	1	0,5	0,47	0,42	0,98	0,94	0,8407	0,666
9	1	1	0,5	0	0,85	0,64	0,63	0,42	0,99	0,94	0,8457	0,694
10	1	1	0,5	0	0,83	0,58	0,63	0,42	0,99	0,94	0,8422	0,682
11	1	1	0,44	0,44	1	0,25	0,47	0,42	0,95	0,94	0,8272	0,66
12	1	1	0,44	0,44	0,71	0,64	0,84	0,42	0,95	0,94	0,8431	0,738

13	1	1	0,44	0,44	0,75	0,58	0,63	0,42	0,99	0,94	0,8187	0,726
14	1	1	0,45	0,45	0,8	0,4	0,73	0,42	0,98	0,93	0,847	0,688
15	1	1	0,45	0,45	0,78	0,64	0,78	0,42	0,98	0,93	0,8568	0,736
16	1	1	0,54	0,45	0,75	0,58	0,73	0,42	0,99	0,93	0,8498	0,724
17	1	1	0,33	0,33	0,78	0,56	0,68	0,42	0,99	0,93	0,8254	0,71
18	1	1	0,5	0	0,875	0,5	0,73	0,42	0,95	0,94	0,8586	0,67
19	1	1	0,44	0,44	0,75	0,63	0,68	0,42	0,98	0,93	0,8278	0,734
20	1	1	0,45	0,45	0,78	0,56	0,78	0,42	0,99	0,93	0,8586	0,722
21	1	1	0,33	0,33	1	0,25	0,63	0,42	0,99	0,93	0,8589	0,646
22	1	1	0,33	0,33	0,71	0,64	0,78	0,42	0,98	0,93	0,831	0,724
23	1	1	0,33	0,33	0,75	0,58	0,68	0,42	0,98	0,93	0,817	0,712
24	1	1	0,5	0	1	0,5	0,63	0,42	0,97	0,94	0,8697	0,666
25	1	1	0,5	0	0,92	0,64	0,52	0,42	0,97	0,94	0,8336	0,694
26	1	1	0,5	0	0,83	0,58	0,63	0,42	0,95	0,94	0,8299	0,682
27	1	1	0,44	0,44	1	0,25	0,42	0,42	0,99	0,94	0,8271	0,66
28	1	1	0,55	0,44	0,78	0,64	0,78	0,42	0,94	0,94	0,8537	0,738
29	1	1	0,44	0,44	0,75	0,58	0,63	0,42	0,98	0,94	0,8168	0,726
30	1	1	0,45	0,45	1	0,4	0,84	0,42	0,99	0,93	0,8726	0,688
31	1	1	0,45	0,45	0,78	0,64	0,73	0,42	0,98	0,93	0,8457	0,736
32	1	1	0,45	0,45	0,83	0,58	0,78	0,42	0,98	0,93	0,8653	0,724

Tabella 6.4: Sintesi dei valori dei parametri normalizzati e delle fitness ottenuti con “GenIAL” e con l’LCR

Si mostrano inoltre i risultati (Figura 6.11) con un istogramma che dimostra graficamente quanto le funzioni di fitness ottenute con l’algoritmo genetico implementato siano superiori a quelle ottenute con il metodo LCR.

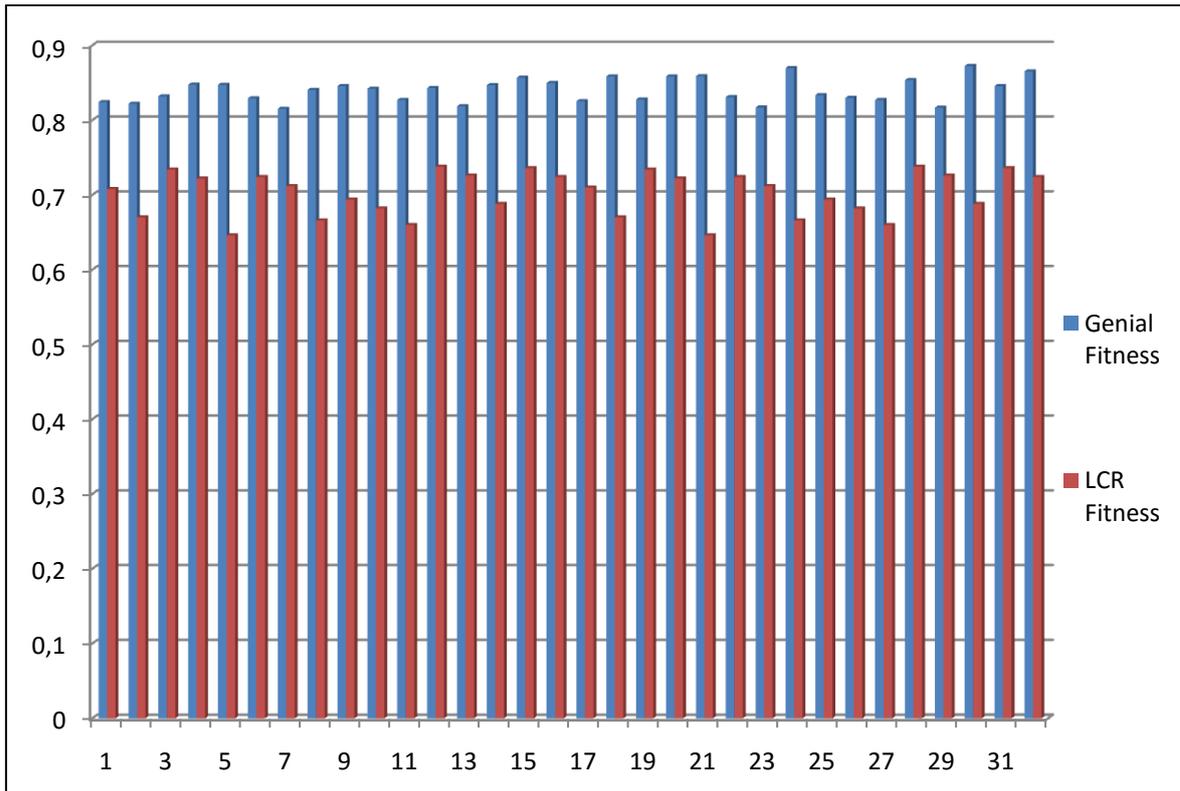


Figura 6.11: Valori delle funzioni di fitness ottenuti con i due metodi utilizzati: “GenIAL” e LCR

La percentuale di miglioramento dal metodo LCR all’algoritmo implementato è del 14%.

Per dimostrare ulteriormente la bontà del metodo implementato, si mostra come “GenIAL” reagisce nel momento in cui si cambiano anche solo singolarmente i valori di alcuni parametri.

L’esempio che segue considera come configurazione (Tabella 6.5) quella che varia mediamente le skill rispetto alla configurazione di partenza (Tabella 6.2).

Operazione	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Direzione
1	0,16	1	1	-	-	-x
2	0,1	0	1	-	-	-y
3	0,14	1	-	-	-	-y
4	0,1	0	-	-	-	-y
5	0,1	1	-	-	-	-y
6	0,34	1	-	-	-	-y
7	0,14	0	1	-	-	-x
8	0,14	0	-	-	-	-x
9	0,16	1	1	-	-	-x
10	0,34	1	2	-	-	-x
11	0,1	1	-	-	-	-z

12	0,67	2	2	-	-	+x
13	0,34	1	4	-	-	+y
14	0,34	1	4	-	-	+y
15	0,1	0	-	-	-	-y
16	0,34	2	4	-	-	+y
17	0,34	1	3	-	-	+y
18	0,1	0	-	-	-	+y
19	0,42	2	-	-	-	+y
20	0,1	1	-	-	-	-y
21	0,85	2	2	-	-	+y
22	0,34	1	3	-	-	+z
23	0,14	1	-	-	-	+y
24	0,34	1	2	-	-	+y
25	1,5	2	5	-	-	-y

Tabella 6.5: Configurazione che varia mediamente le skill rispetto a quella di partenza

La miglior soluzione che si ottiene con la configurazione precedente è rappresentata in Figura 6.12, dove si presentano le caratteristiche di ciascuna stazione.

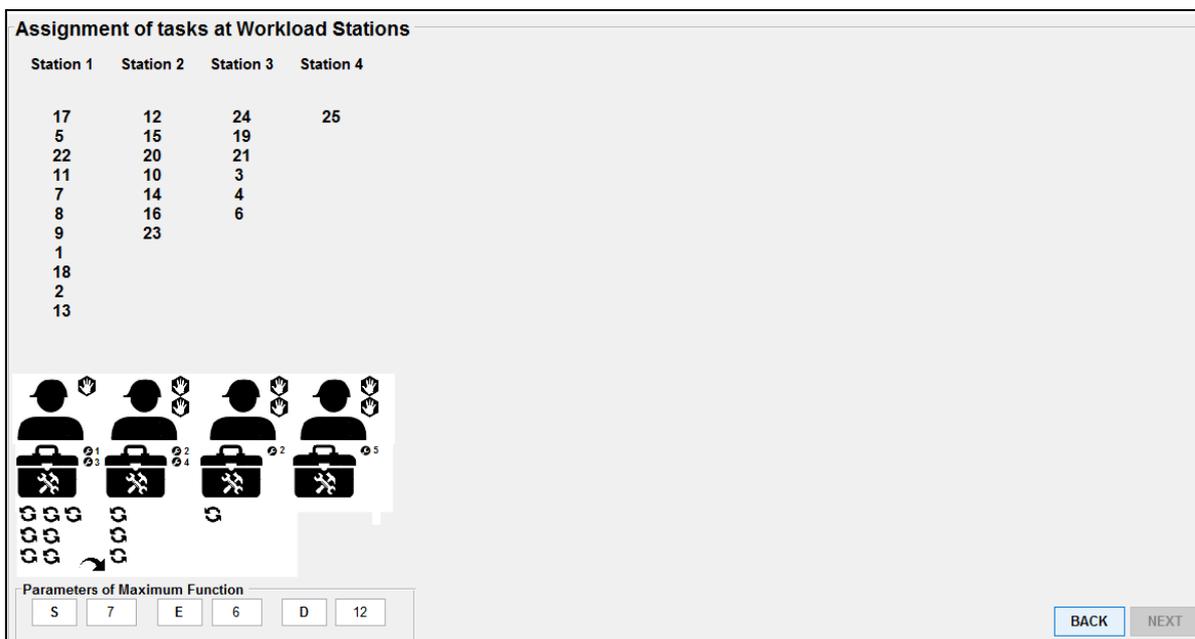


Figura 6.12: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione di lavoro (configurazione Tabella 6.4)

Se si prova a variare la skill di un'operazione elementare, si nota come "GenIAL" ottiene una soluzione differente. Nel dettaglio, si varia la skill dell'operazione 19, da 2 a 1. La soluzione migliore ottenuta in Figura 6.13.

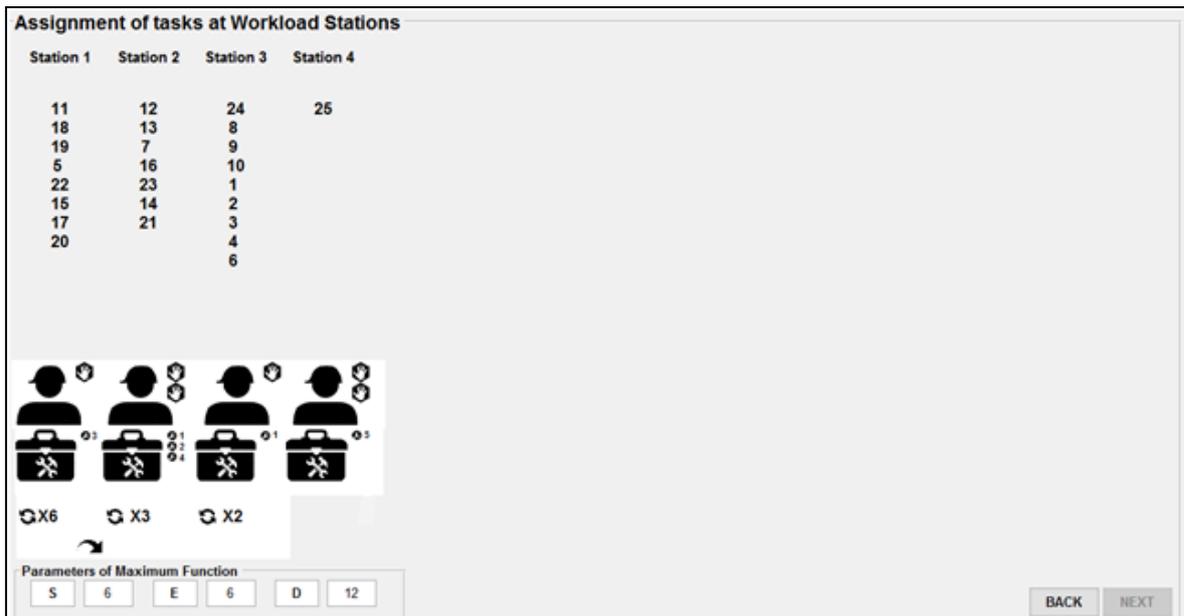


Figura 6.13: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione di lavoro (configurazione Tabella 6.4 con variazione skill operazione 19)

Le differenze fra Figura 6.12 e Figura 6.13 mostrano che l'operazione 19 è stata spostata dalla stazione 3 alla stazione 1, e questo ha permesso di individuare una soluzione che richiede in totale due stazioni invece che tre con skill massima (pari a 2). Si può notare infatti che l'operazione 19 prima veniva eseguita insieme ad altre operazioni che richiedevano una skill pari a 2 come la 21; adesso invece viene raggruppata in una stazione dove tutte le operazioni richiedono skill  $\leq 1$ , e la 21 in una dove ci sono altre operazioni (ad esempio la 12 e la 16) che necessitano skill pari a 2.

## 6.2 Caso di studio: Tonge

Il secondo caso di studio analizzato si è individuato in letteratura, ed è uno dei casi più utilizzati dai ricercatori come punto di riferimento per poter confrontare le tecniche implementate (euristiche o iterative), e valutare eventuali miglioramenti alla tecnica ideata dall'autore Fred. M Tonge.

### 6.2.1 Descrizione

Il problema in esame è un prodotto dell'industria elettronica (37) costituito da 70 operazioni elementari delle quali sono noti i tempi di esecuzione e il tempo ciclo (in secondi) utilizzati dall'autore. Di seguito si riporta il grafo delle precedenze di Tonge (37).

Parametri del problema, come abilità ed attrezzature richieste o direzioni di assemblaggio sono state ipotizzate in quanto oltre al tempo di esecuzione delle operazioni, alle precedenze e al tempo ciclo non si hanno altre informazioni a disposizione sul presente

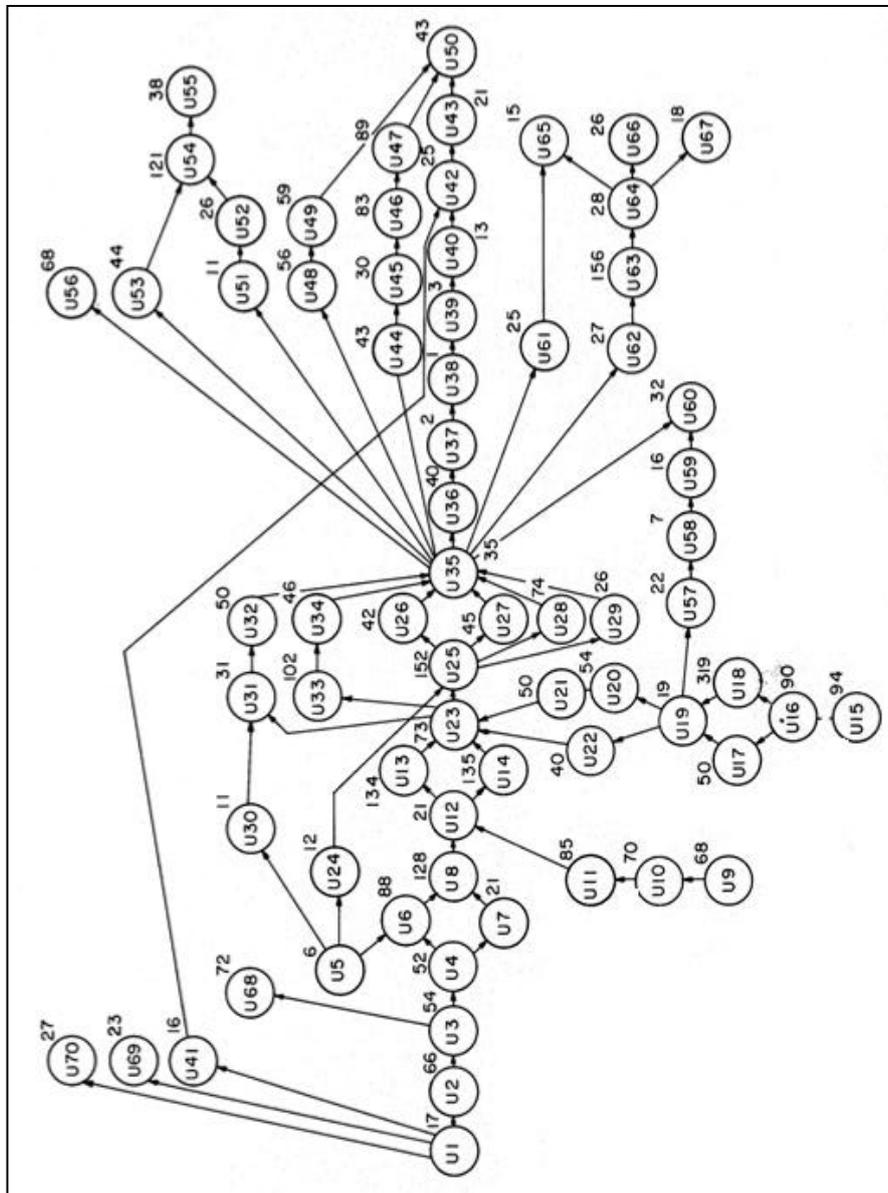


Figura 6.14: Grafo delle precedenze di Tonge

caso di studio.

### 6.2.2 Implementazione in “GenIAL”

La prima configurazione che si analizza è rappresentata in Tabella 6.6.

Operazione	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Direzione
1	0,28	0	-	-	-	-y
2	1,1	1	1	-	-	-y
3	0,9	0	-	-	-	-x
4	0,86	1	1	-	-	-y
5	0,1	0	-	-	-	+x
6	1,46	1	1	2	-	-y
7	0,35	0	-	-	-	+y
8	2,1	2	2	3	-	-y
9	1,13	1	1	-	-	-y
10	1,16	1	3	-	-	+y
11	1,4	1	2	-	-	+y
12	0,35	0	-	-	-	+z
13	2,23	1	3	4	-	-y
14	2,25	1	4	-	-	-y
15	1,56	0	-	-	-	+x
16	1,5	1	1	-	-	-y
17	0,83	0	-	-	-	-y
18	1,98	2	2	3	4	-y
19	0,31	0	-	-	-	-x
20	0,9	0	-	-	-	-x
21	0,83	0	1	-	-	-y
22	0,66	0	-	-	-	-y
23	1,21	2	1	3	-	-y
24	0,2	0	-	-	-	-z
25	2,53	2	1	5	-	-y
26	0,7	0	-	-	-	-y
27	0,75	1	4	-	-	-x
28	1,23	1	3	-	-	-x
29	0,43	0	-	-	-	-y
30	0,18	0	-	-	-	-y
31	0,51	1	1	-	-	-y
32	0,83	0	-	-	-	+y
33	1,7	1	2	5	-	-y
34	0,76	0	-	-	-	+x
35	0,58	1	3	6	-	-y
36	0,66	0	-	-	-	+x
37	0,03	0	-	-	-	+x

38	0,01	0	-	-	-	+z
39	0,05	0	-	-	-	+z
40	0,21	0	-	-	-	+z
41	0,26	0	-	-	-	+z
42	0,41	0	-	-	-	+x
43	0,35	1	3	-	-	-y
44	0,71	1	4	5	-	-y
45	0,5	0	-	-	-	-z
46	1,38	2	5	7	-	-y
47	1,48	2	7	8	-	-y
48	0,93	0	2	-	-	-y
49	0,98	1	-	-	-	+y
50	0,71	0	3	-	-	+z
51	0,18	0	-	-	-	+y
52	0,43	0	-	-	-	+y
53	0,73	0	4	-	-	-z
54	2,01	2	5	7	8	-y
55	0,63	0	-	-	-	+y
56	1,13	1	3	4	-	+x
57	0,36	0	-	-	-	+y
58	0,11	0	-	-	-	+y
59	0,26	0	-	-	-	+y
60	0,53	1	6	-	-	-x
61	0,41	0	-	-	-	+y
62	0,45	0	-	-	-	+y
63	2,6	2	1	5	8	-y
64	0,46	0	1	-	-	-y
65	0,25	0	-	-	-	+y
66	0,43	0	-	-	-	-y
67	0,3	0	-	-	-	-y
68	1,2	1	5	6	-	-y
69	0,38	0	-	-	-	-x
70	0,45	1	9	-	-	-y

Tabella 6.6: Prima configurazione del caso di studio Tonge

Oltre alla configurazione appena descritta, si sono svolte prove su altrettante configurazioni, create variando i parametri  $S$  ed  $E$  nello stesso modo con cui si sono realizzate per il precedente caso di studio, ovvero:

- a partire dalla configurazione base di  $S$  ed  $E$  si varia per primo  $S$  lasciando inalterati gli altri parametri. Si varia  $S$  per tre volte, e lo si fa partendo dalla minimizzazione delle skill nel primo caso fino alla loro massimizzazione del terzo caso;

- stesso procedimento per gli equipment  $E$ , sempre a partire dalla configurazione descritta prima;
- ad ognuna delle tre configurazioni create con le variazioni di  $S$ , si associano le tre configurazioni create con le variazioni di  $E$ .

Si ottengono quindi 16 configurazioni differenti da analizzare con “GenIAL”.

Si riportano nel prossimo paragrafo i risultati di alcune configurazioni, quelli considerati più rilevanti; si allegano invece i risultati della maggior parte delle configurazioni in Appendice A.

### 6.2.3 Risultati

Si riporta come primo esempio il miglior risultato ottenuto per la configurazione che massimizza gli equipment rispetto alla configurazione di partenza. Si mostra in Figura 6.15 l’interfaccia grafica che rappresenta l’output della miglior soluzione e in Figura 6.16 e 6.17 quelle che dettagliano le caratteristiche di ciascuna stazione di lavoro.

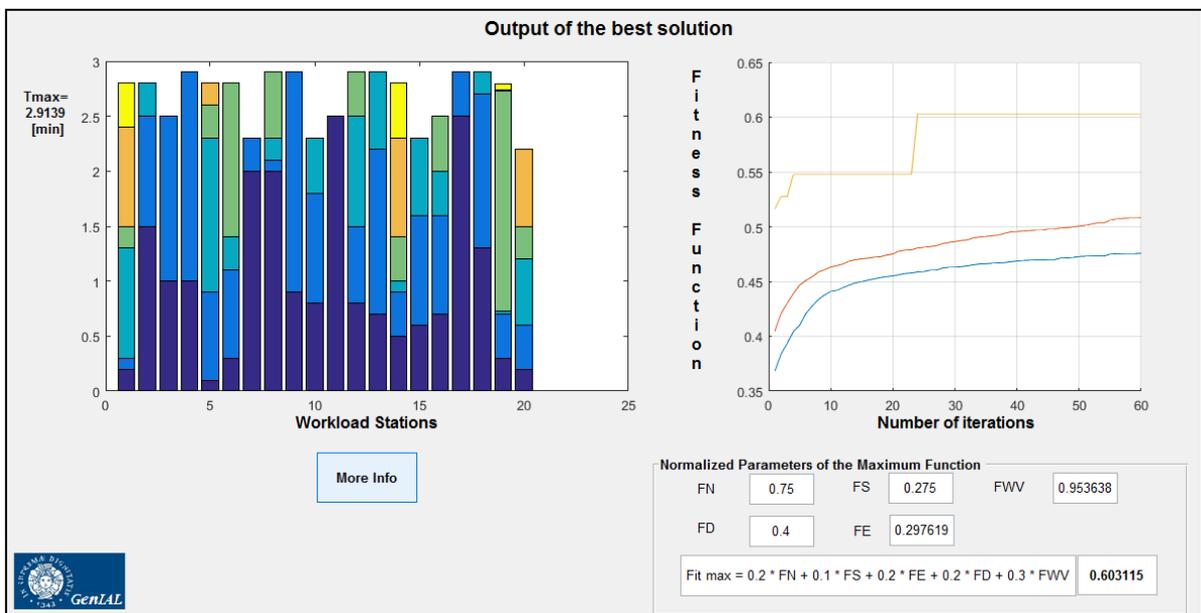


Figura 6.15: Miglior soluzione per la configurazione che massimizza gli equipment rispetto alla configurazione di partenza

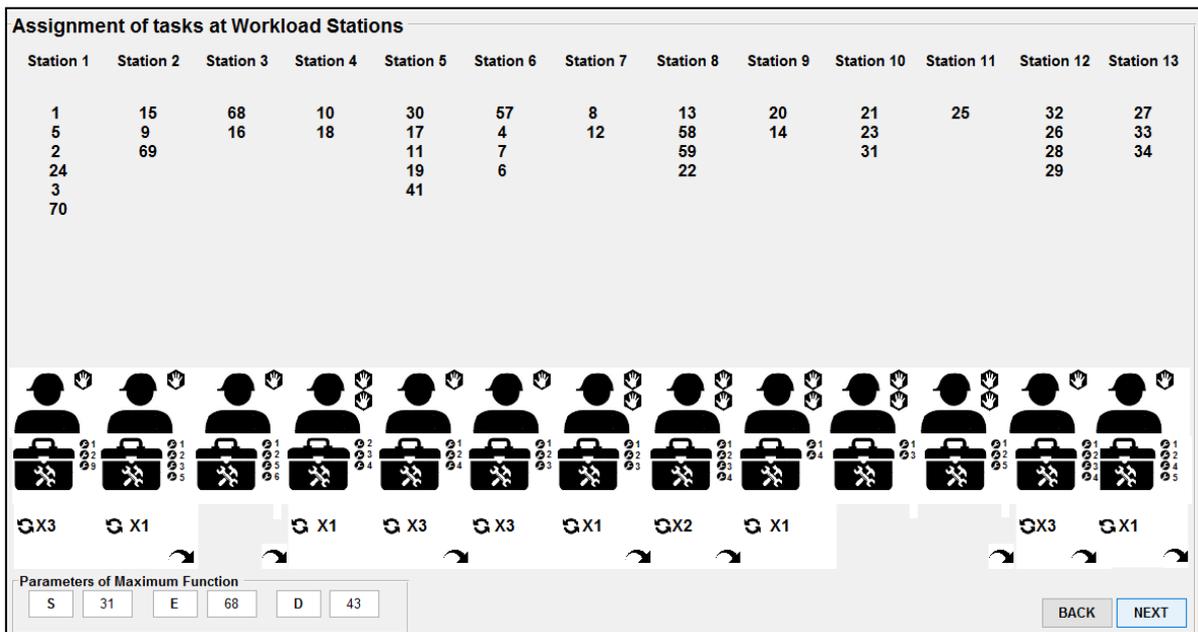


Figura 6.16: Interfaccia grafica che dettaglia le caratteristiche delle stazioni di lavoro per la configurazione che massimizza gli equipment rispetto alla configurazione di partenza (dalla 1° alla 13°)

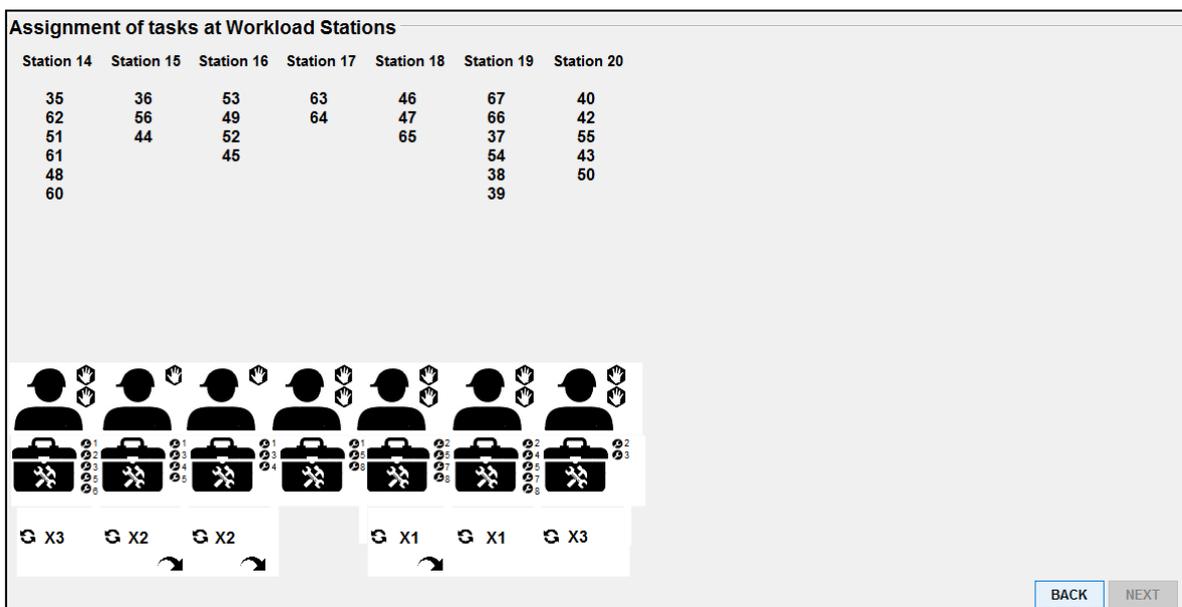


Figura 6.17: Interfaccia grafica che dettaglia le caratteristiche delle stazioni di lavoro per la configurazione che massimizza gli equipment rispetto alla configurazione di partenza (dalla 14° alla 20°)

Per sottolineare la bontà della soluzione individuata da “GenIAL” rispetto a quella che si può ottenere con un metodo euristico, si risolve la stessa configurazione con il metodo euristico LCR e si ottiene il seguente risultato (Figura 6.18).

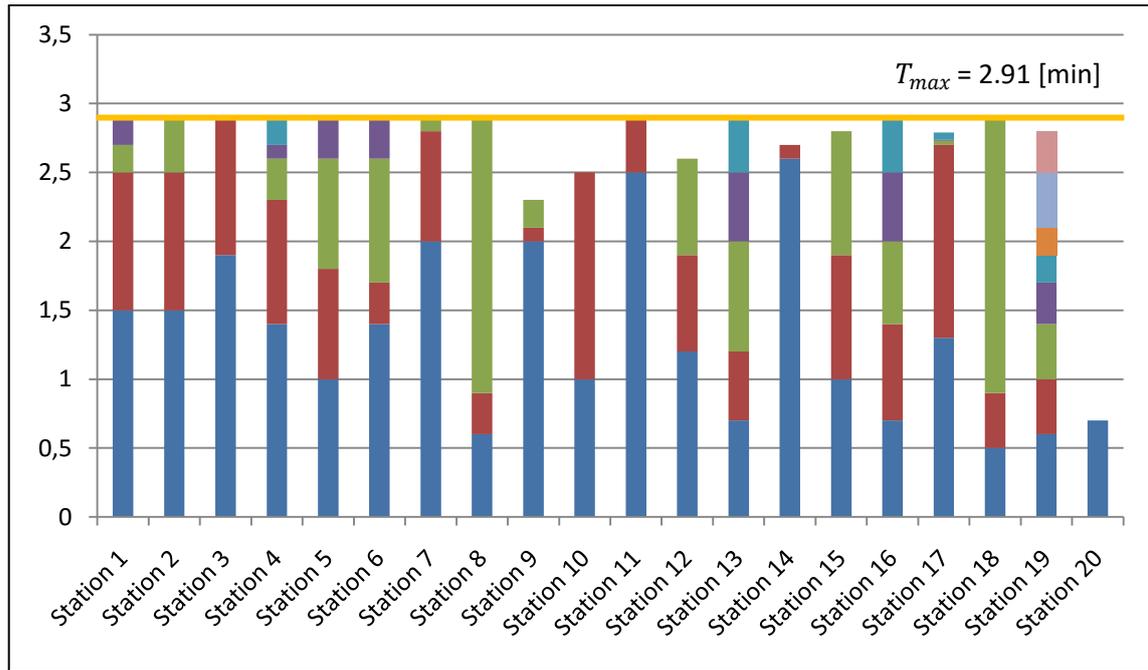


Figura 6.18: Istogramma della soluzione con metodo LCR per la configurazione che massimizza gli equipment rispetto alla configurazione di partenza

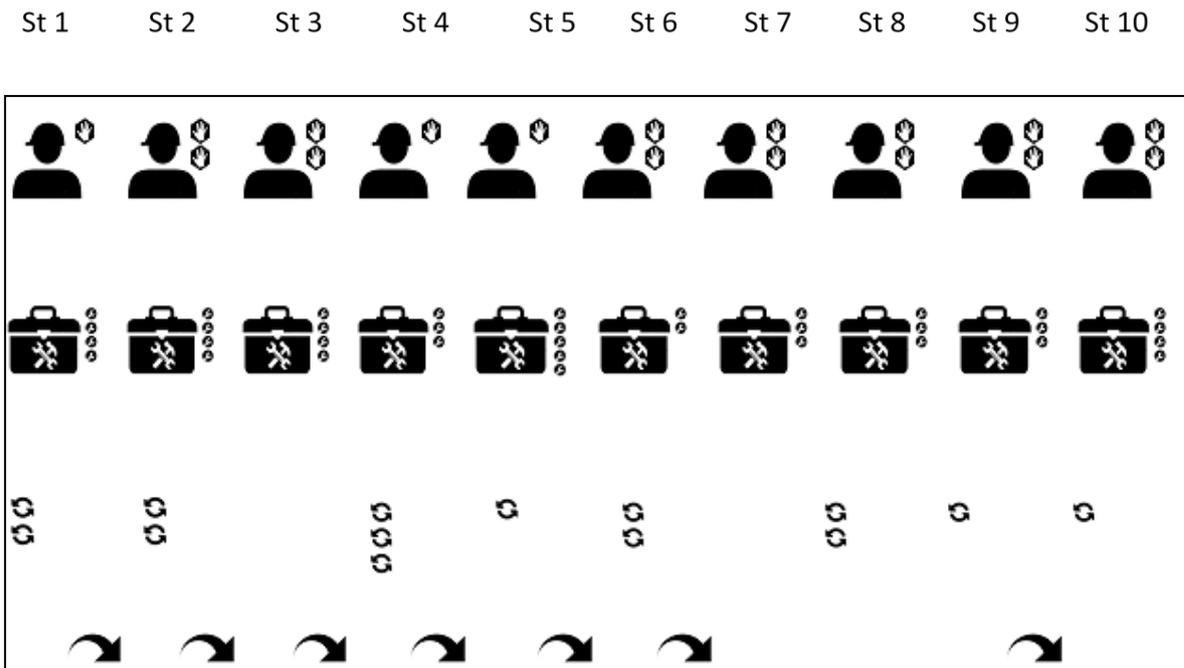


Figura 6.19: : Caratteristiche di ciascuna stazione di lavoro ottenute con il metodo LCR per la configurazione che massimizza gli equipment rispetto alla configurazione di partenza (dalla 1° alla 10° stazione)

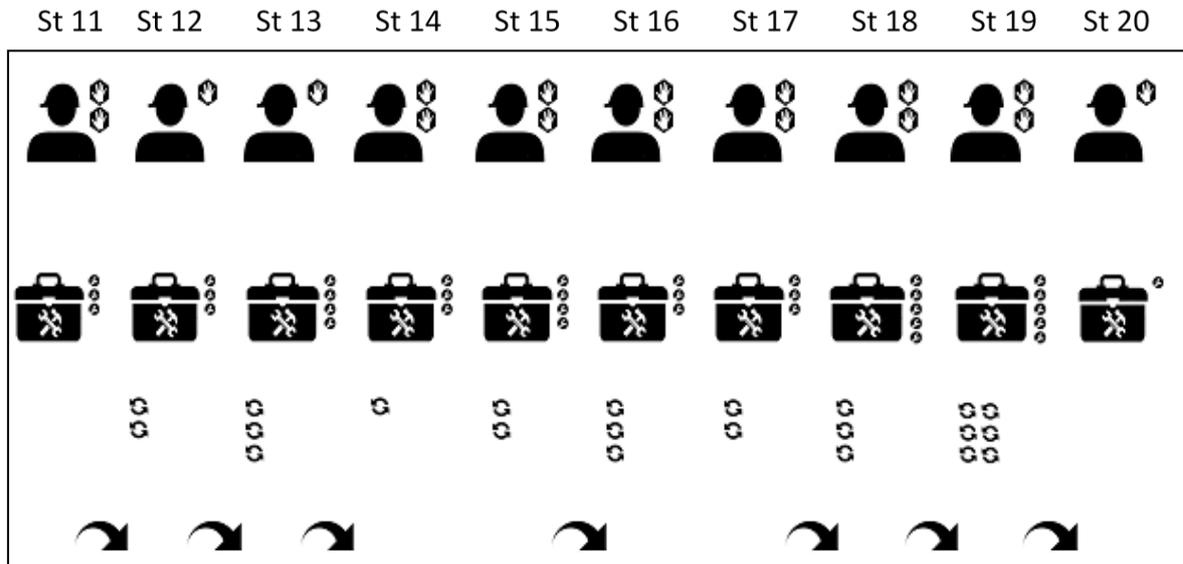


Figura 6.20: : Caratteristiche di ciascuna stazione di lavoro ottenute con il metodo LCR per la configurazione che massimizza gli equipment rispetto alla configurazione di partenza (dalla 11° alla 20° stazione)

Come si nota dalle differenza fra le Figure 6.16 e 6.17, e 6.19 e 6.20, la soluzione con “GenIAL” richiede in totale 31 skill, 68 equipment e 43 cambi di direzione mentre la soluzione con LCR richiede in totale 34 skill, 68 equipment e 51 cambi di direzione.

Si calcola infine la funzione di fitness che si ottiene con la soluzione del metodo LCR:

- $FN = 0.75$ ;
- $FS = 0.2$ ;
- $FD = 0.28$ ;
- $FE = 0.29$ ;
- $FWV = 0.82$ ;

$$F = 0.2 * FN + 0.1 * FS + 0.2 * FD + 0.2 * FE + 0.3 * FWV = 0.53$$

invece di 0.6 ottenuto con l’algoritmo implementato.

Il secondo esempio riguarda la miglior soluzione che si ottiene per la configurazione che minimizza le skill e varia mediamente gli equipment rispetto alla configurazione di partenza; si mostrano in Figura 6.21, 6.22 e 6.23 i risultati ottenuti.

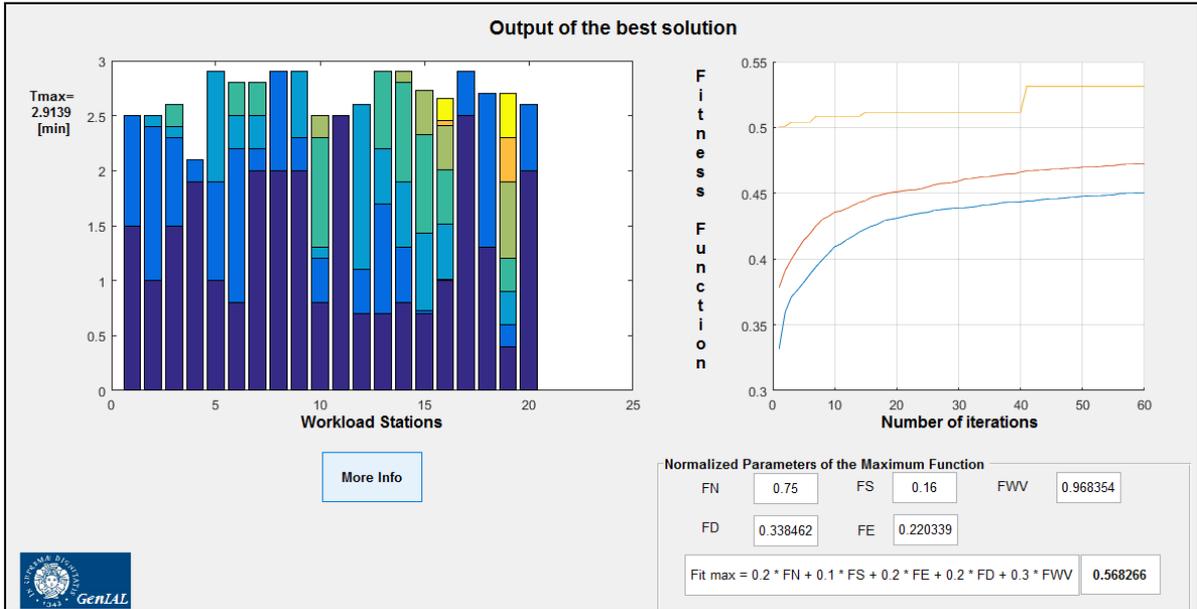


Figura 6.21: Miglior soluzione per la configurazione che minimizza le skill e varia mediamente gli equipment rispetto alla configurazione di partenza

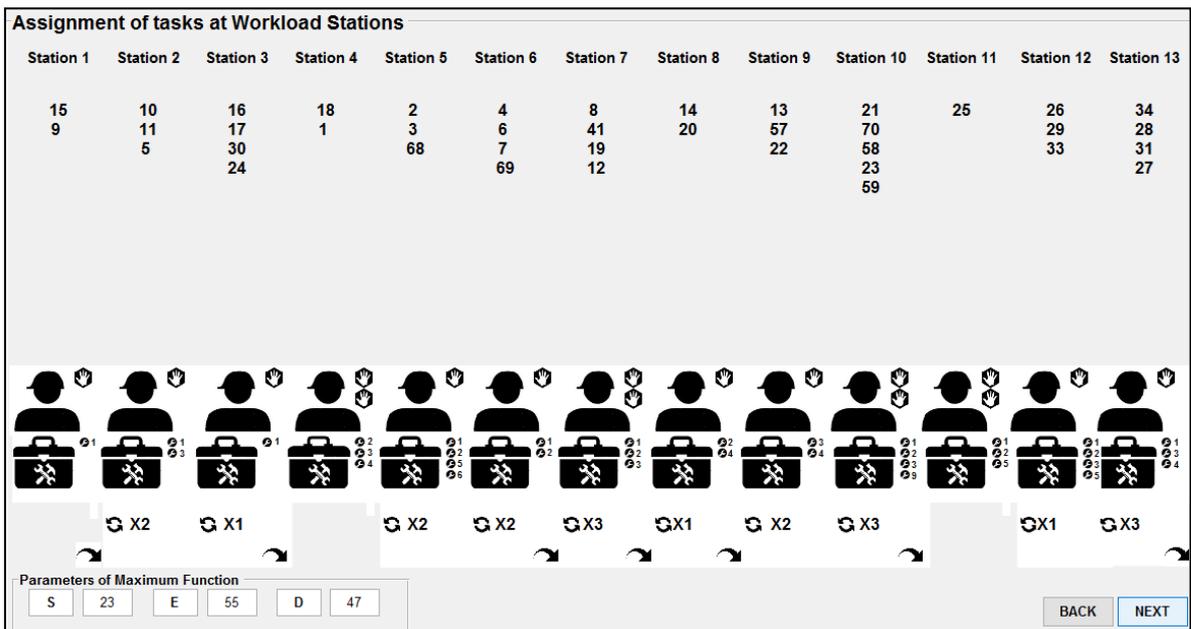


Figura 6.22: Interfaccia grafica che dettaglia le caratteristiche delle stazioni di lavoro soluzione per la configurazione che minimizza le skill e varia mediamente gli equipment rispetto alla configurazione di partenza (dalla 1° alla 13°)

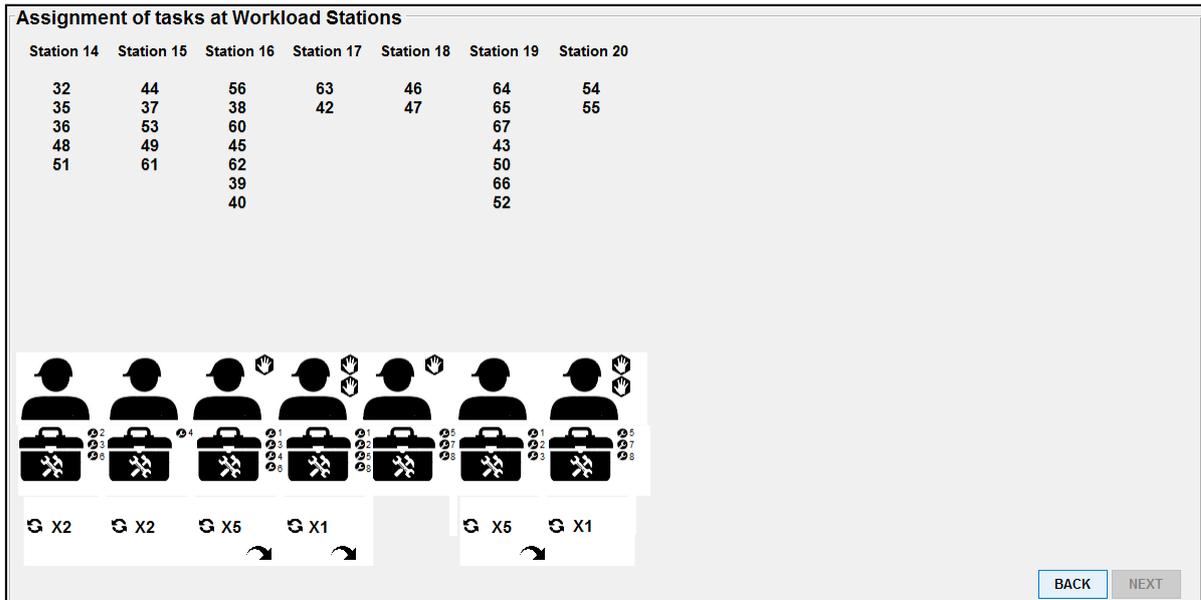


Figura 6.23: Interfaccia grafica che dettaglia le caratteristiche delle stazioni di lavoro soluzione per la configurazione che minimizza le skill e varia mediamente gli equipment rispetto alla configurazione di partenza (dalla 14° alla 20°)

Anche questa configurazione si risolve con il metodo euristico LCR e si ottengono le seguenti differenze:

- 23 skill con “GenIAL”, 22 con LCR;
- 55 equipment con “GenIAL” e 56 con LCR;
- 47 cambi di direzione con “GenIAL” e 49 con LCR.

Si calcola la funzione di fitness della soluzione individuata con LCR:

- $FN = 0.75$ ;
- $FS = 0.2$ ;
- $FD = 0.31$ ;
- $FE = 0.2$ ;
- $FWV = 0.82$ ;

$$F = 0.2 * FN + 0.1 * FS + 0.2 * FD + 0.2 * FE + 0.3 * FWV = 0.52$$

invece di 0.56 ottenuto con l’algoritmo implementato.

Come già anticipato, il caso di studio Tonge proviene dalla letteratura ed è utilizzato dagli studiosi per confrontare i risultati ottenuti con quelli con cui l'autore Fred M. Tonge ha per primo svolto il caso.

Nell'articolo del 1960 di Tonge (38), il caso è risolto attraverso un metodo e la soluzione prevede, con un tempo ciclo pari a 2.93 min, una linea composta da 22 stazioni di lavoro. Thomas R. Hoffmann (39) ha risolto successivamente il caso Tonge con un differente metodo e, con lo stesso tempo ciclo, ha individuato un soluzione con una linea composta da 21 stazioni.

Nel 2011, O. Kilincci (37) propone un confronto fra diversi metodi che impiegano differenti algoritmi genetici per risolvere il caso Tonge, utilizzando 13 tempi ciclo differenti. Facendo riferimento al tempo ciclo del presente lavoro, si presentano le soluzioni ottenute:

- metodo *Moodie & Young*: soluzione con 22 stazioni;
- metodo *Tonge MIF*: soluzione con 23 stazioni;
- metodo *Tonge RC*: soluzione con 23 stazioni;
- metodo *Tonge BPC*: soluzione con 21 stazioni;
- metodo *Nevins BBS*: soluzione con 21 stazioni;
- metodo *Baybars LBHA-1*: soluzione con 22 stazioni;
- metodo *FSb*: soluzione con 22 stazioni.

Confrontando le soluzioni dei metodi appena proposti con “GenIAL”, si evince che l'algoritmo implementato nel presente lavoro di tesi è in grado di individuare soluzioni migliori in termini di numero di stazioni di lavoro: durante lo studio delle 16 configurazioni analizzate per Tonge, 4 soluzioni richiedono 21 stazioni e le restanti richiedono 20 stazioni.

### 6.3 Conclusioni

L'analisi dei risultati ottenuti sui due casi di studio appena descritti è in grado di mostrare la bontà dell'algoritmo genetico implementato nel presente lavoro di tesi.

Il primo caso di studio (Carburetor Keihin 19 PD), caso reale di assemblaggio di un carburatore, è caratterizzato da elevati valori delle funzioni di fitness per ciascuna

soluzione, e soprattutto valori migliori rispetto alle soluzioni ottenute applicando il metodo euristico LCR alle stesse configurazioni.

Oltre alle fitness ottenute però, con questo caso di studio si comprendono le capacità che “GenIAL” dimostra nel riuscire ad ottimizzare gli obiettivi da raggiungere; il carattere innovativo del presente algoritmo si evidenzia soprattutto nella distribuzione di skill, equipment e direzioni di assemblaggio fra le stazioni di lavoro e nella sua capacità di minimizzarli. Quindi, diversamente da quanto presente nell’80% della letteratura riguardo agli algoritmi genetici, non si cerca soltanto di minimizzare il numero di stazioni e la varianza del carico di lavoro fra le stazioni, ma si introducono nuovi obiettivi che si ritengono necessari ed efficaci per una efficiente gestione del sistema produttivo, permettendo di ottenere risultati migliori, come visto nel paragrafo 6.1.3.

Il secondo caso di studio (Tonge), ci consente di confrontare “GenIAL” con i vari metodi utilizzati negli anni per risolverlo. A partire dal metodo LCR impiegato come paragone in questo lavoro, al metodo euristico dell’autore Tonge, a tutti gli strumenti utilizzati in letteratura, “GenIAL” è in grado di fornire soluzioni che sono uguali o migliori nella maggior parte dei casi in termini di stazioni di lavoro necessarie nella linea di montaggio.

Si propone infine la possibilità di studi e sviluppi futuri sul lavoro svolto, rispetto a più punti di vista:

- nei casi di studio analizzati, non sono stati valutati a 360° tutte le possibili variazioni applicabili all’algoritmo: si potrebbe pensare per esempio di variare non solo i parametri  $S$ ,  $E$  e  $D$ , ma anche i parametri dell’algoritmo (numero iterazioni, numero individui, frazione popolazione e probabilità degli operatori genetici) ed esaminare varie combinazioni di questi; oppure si potrebbero modificare i pesi attribuiti a ciascun obiettivo nella funzione di fitness;
- confronto di “GenIAL” con più casi di studio presenti in letteratura, anche maggiormente complessi rispetto a quelli presentati;
- miglioramento dell’interfaccia utente, sia per l’inserimento dei dati, che per la visualizzazione dell’output.

## Appendice A

Si descrivono nella presente appendice per prima i risultati ottenuti nell'applicazione dell'algoritmo implementato ad alcune configurazioni del primo caso di studio: Carburetor Keihin 19 PD. Nella seconda parte si presentano invece i risultati di alcune configurazioni del secondo caso di studio: Tonge.

### Caso: Carburetor Keihin 19 PD

#### 1. Configurazione che varia mediamente le skill rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,16	1	1	0	0	-x
2	0,1	0	1	0	0	-y
3	0,14	1	0	0	0	-y
4	0,1	0	0	0	0	-y
5	0,1	1	0	0	0	-y
6	0,34	1	0	0	0	-y
7	0,14	0	1	0	0	-x
8	0,14	0	0	0	0	-x
9	0,16	1	1	0	0	-x
10	0,34	1	2	0	0	-x
11	0,1	1	0	0	0	-z
12	0,67	2	2	0	0	+x
13	0,34	1	0	0	0	+y
14	0,34	1	4	0	0	+y
15	0,1	0	0	0	0	-y
16	0,34	2	4	0	0	+y
17	0,34	1	3	0	0	+y
18	0,1	0	0	0	0	+y
19	0,42	2	0	0	0	+y
20	0,1	1	0	0	0	-y
21	0,1	2	2	0	0	+y
22	0,34	1	3	0	0	+z
23	0,14	1	0	0	0	+y
24	0,34	1	2	0	0	+y
25	1,5	2	5	0	0	-y

Tabella A.1: Configurazione che varia mediamente le skill rispetto a quella di partenza

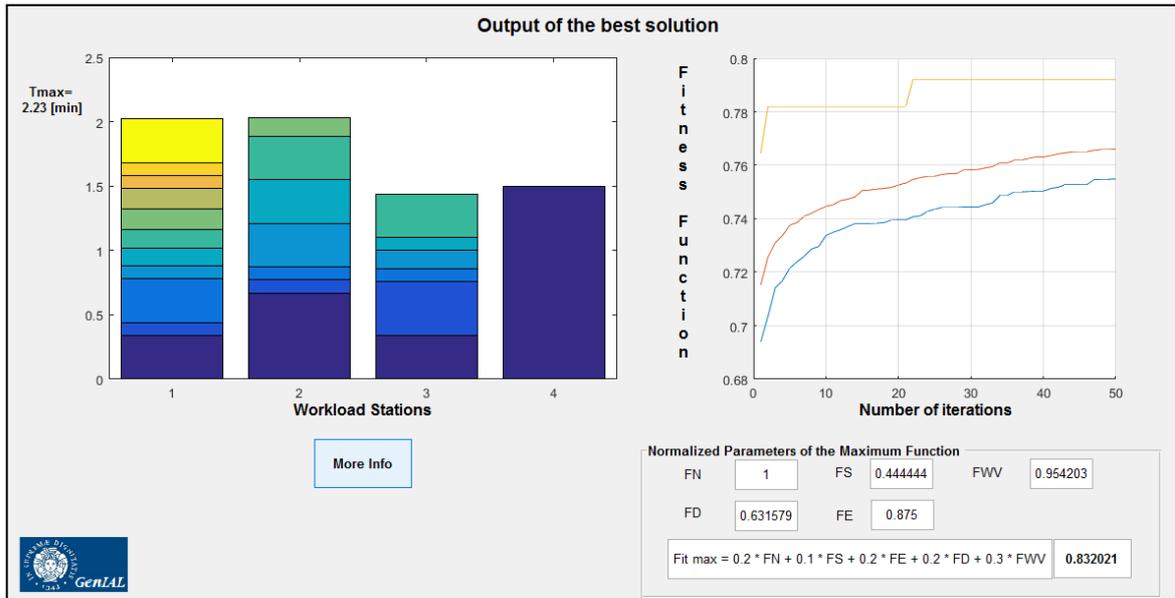


Figura A.1: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che varia mediamente le skill rispetto a quella di partenza

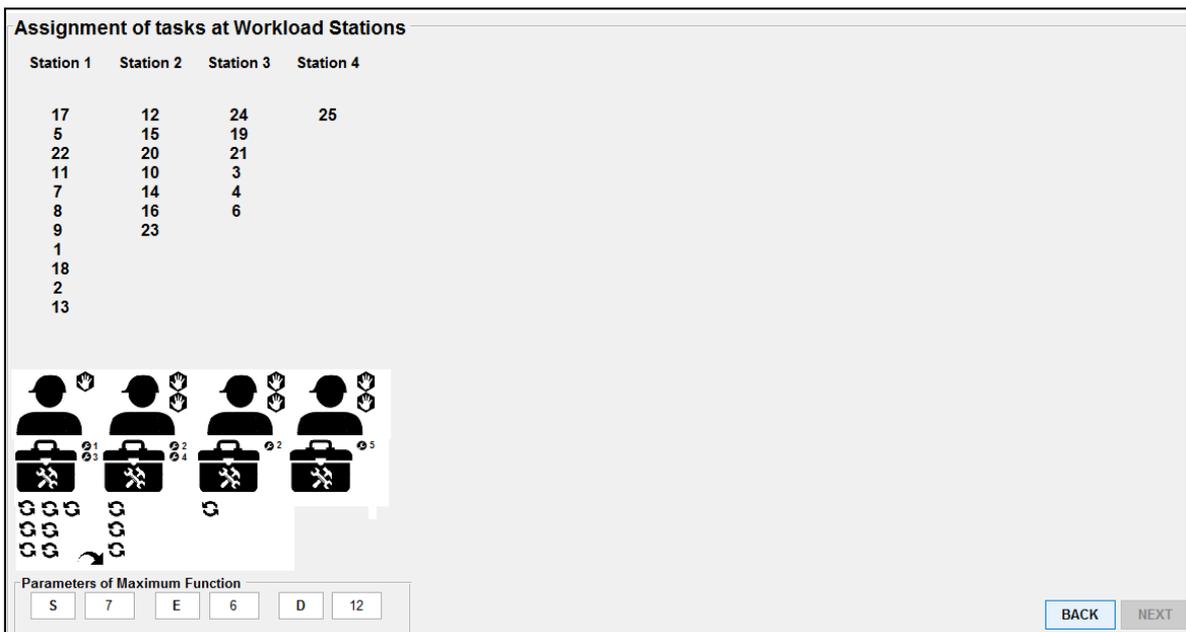


Figura A.2: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che varia mediamente le skill rispetto a quella di partenza

2. Configurazione che massimizza le skill rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,16	1	1	0	0	-x
2	0,1	2	1	0	0	-y
3	0,14	1	0	0	0	-y
4	0,1	0	0	0	0	-y
5	0,1	1	0	0	0	-y
6	0,34	1	0	0	0	-y
7	0,14	0	1	0	0	-x
8	0,14	0	0	0	0	-x
9	0,16	1	1	0	0	-x
10	0,34	2	2	0	0	-x
11	0,1	0	0	0	0	-z
12	0,67	1	2	0	0	+x
13	0,34	1	4	0	0	+y
14	0,34	2	4	0	0	+y
15	0,1	0	0	0	0	-y
16	0,34	2	4	0	0	+y
17	0,34	2	3	0	0	+y
18	0,1	0	0	0	0	+y
19	0,42	1	0	0	0	+y
20	0,1	1	0	0	0	-y
21	0,85	2	2	0	0	+y
22	0,34	1	3	0	0	+z
23	0,14	1	0	0	0	+y
24	0,34	1	2	0	0	+y
25	1,5	2	5	0	0	-y

Tabella A.2: Configurazione che massimizza le skill rispetto a quella di partenza

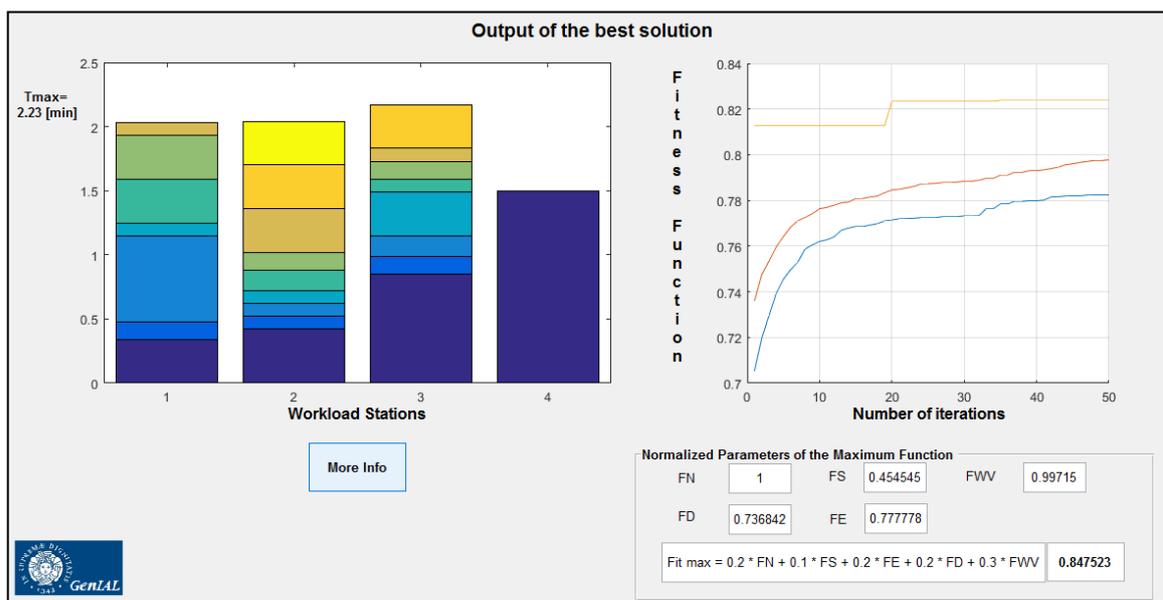


Figura A.3: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che massimizza le skill rispetto a quella di partenza

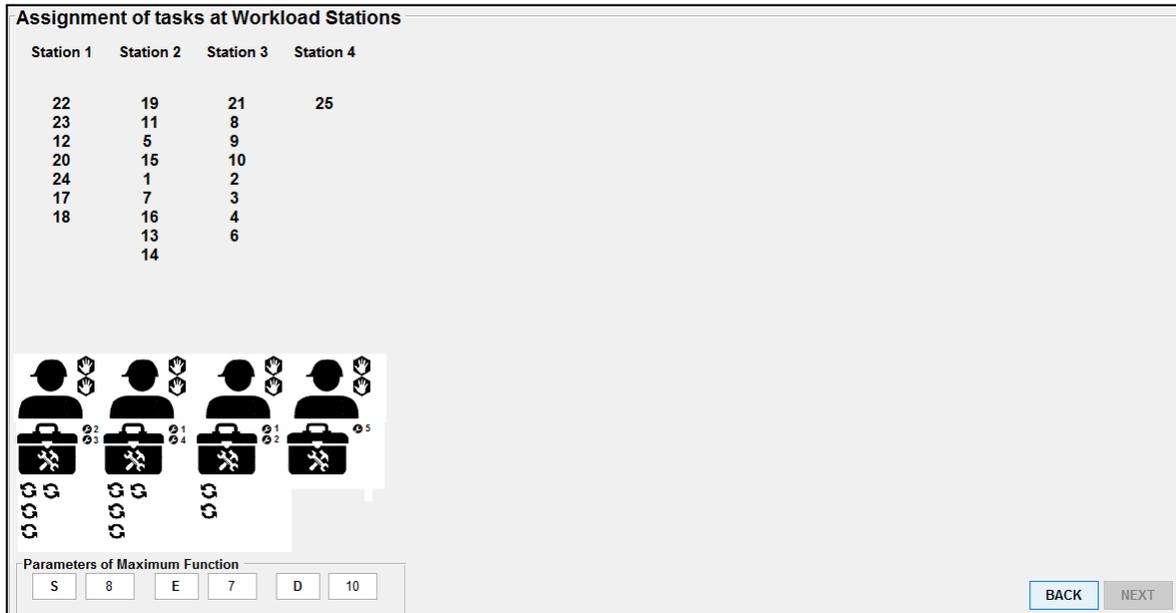


Figura A.4: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che massimizza le skill rispetto a quella di partenza

3. Configurazione che minimizza gli equipment rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,16	1	0	0	0	-x
2	0,1	0	0	0	0	-y
3	0,14	0	0	0	0	-y
4	0,1	0	0	0	0	-y
5	0,1	0	0	0	0	-y
6	0,34	0	0	0	0	-y
7	0,14	0	0	0	0	-x
8	0,14	0	0	0	0	-x
9	0,16	1	0	0	0	-x
10	0,34	1	1	0	0	-x
11	0,1	0	0	0	0	-z
12	0,67	1	1	0	0	+x
13	0,34	1	0	0	0	+y
14	0,34	1	0	0	0	+y
15	0,1	0	0	0	0	-y
16	0,34	1	0	0	0	+y
17	0,34	1	2	0	0	+y
18	0,1	0	0	0	0	+y
19	0,42	1	0	0	0	+y
20	0,1	0	0	0	0	-y
21	0,85	2	1	0	0	+y
22	0,34	1	2	0	0	+z
23	0,14	0	0	0	0	+y
24	0,34	1	2	0	0	+y
25	1,5	2	3	0	0	-y

Tabella A.3: Configurazione che minimizza gli equipment rispetto a quella di partenza

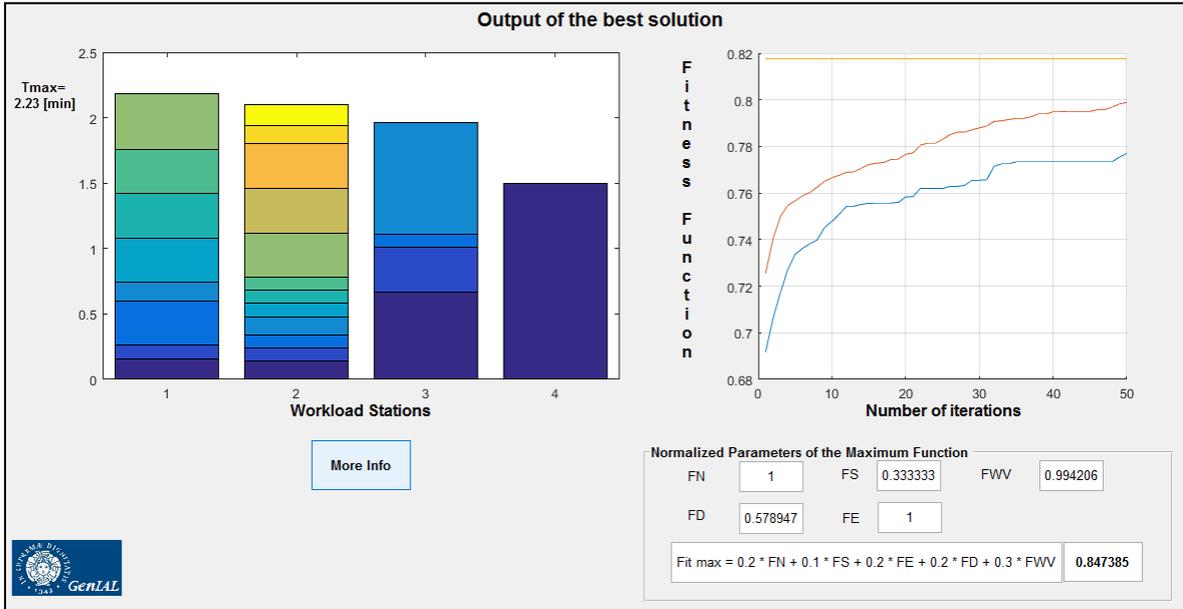


Figura A.5: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che minimizza gli equipment rispetto a quella di partenza

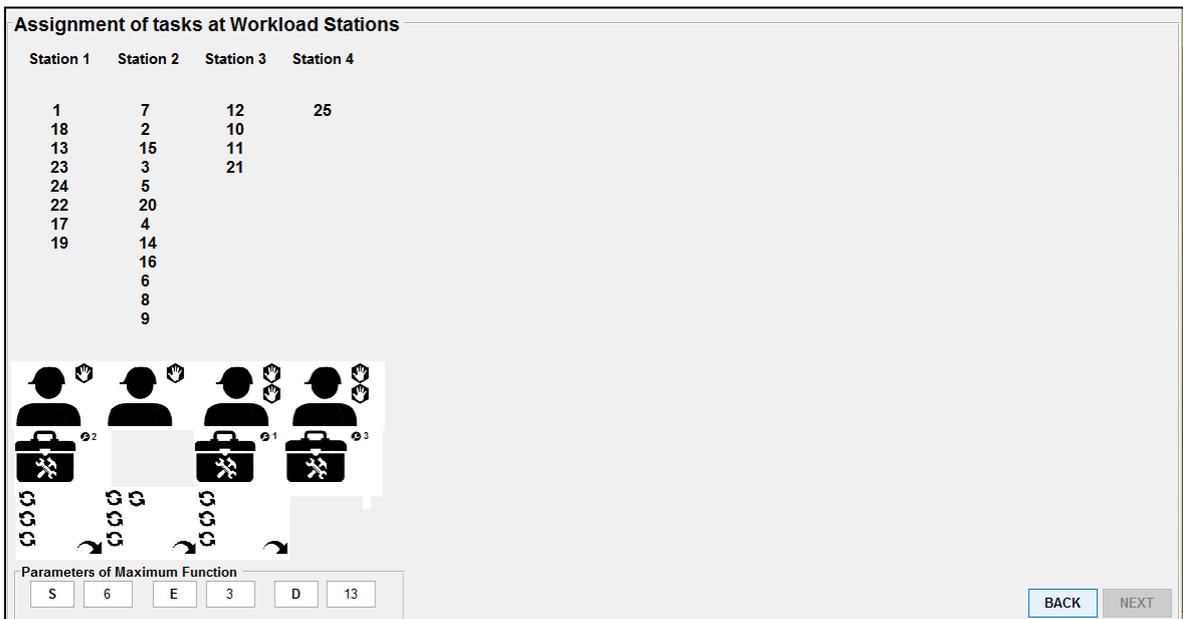


Figura A.6: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che minimizza gli equipment rispetto a quella di partenza

4. Configurazione che varia mediamente gli equipment rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,16	1	1	0	0	-x
2	0,1	0	1	0	0	-y
3	0,14	0	0	0	0	-y
4	0,1	0	0	0	0	-y
5	0,1	0	1	0	0	-y
6	0,34	0	0	0	0	-y
7	0,14	0	1	0	0	-x
8	0,14	0	0	0	0	-x
9	0,16	1	1	0	0	-x
10	0,34	1	2	0	0	-x
11	0,1	0	1	0	0	-z
12	0,67	1	2	0	0	+x
13	0,34	1	4	0	0	+y
14	0,34	1	4	0	0	+y
15	0,1	0	1	0	0	-y
16	0,34	1	4	0	0	+y
17	0,34	1	3	0	0	+y
18	0,1	0	0	0	0	+y
19	0,42	1	0	0	0	+y
20	0,1	0	1	0	0	-y
21	0,85	2	2	0	0	+y
22	0,34	1	3	0	0	+z
23	0,14	0	1	0	0	+y
24	0,34	1	2	0	0	+y
25	1,5	2	5	0	0	-y

Tabella A.4: Configurazione che varia mediamente gli equipment rispetto a quella di partenza

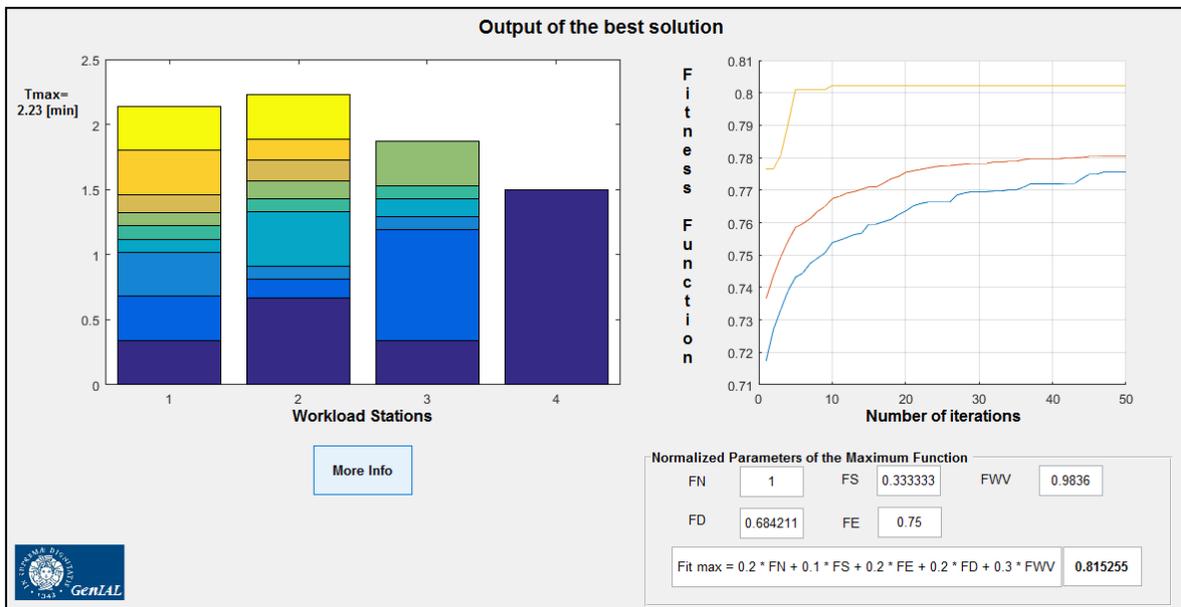


Figura A.7: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che varia mediamente gli equipment rispetto a quella di partenza

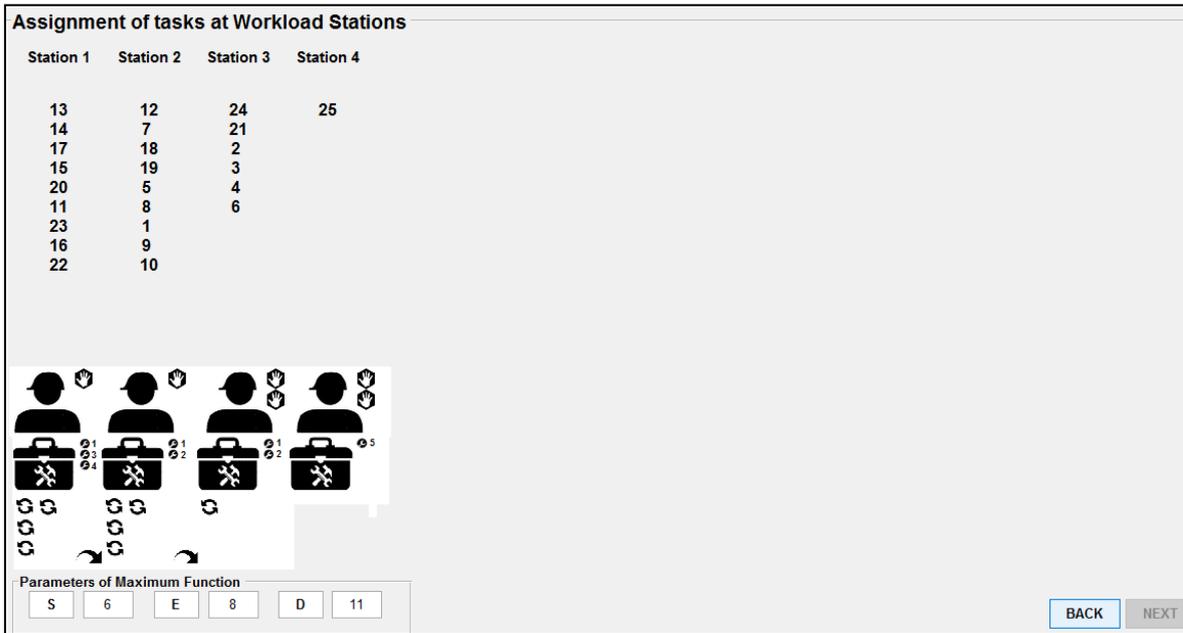


Figura A.8: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che varia mediamente gli equipment rispetto a quella di partenza

5. Configurazione che massimizza gli equipment rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,16	1	1	0	0	-x
2	0,1	0	1	0	0	-y
3	0,14	0	1	0	0	-y
4	0,1	0	1	0	0	-y
5	0,1	0	0	0	0	-y
6	0,34	0	0	0	0	-y
7	0,14	0	1	0	0	-x
8	0,14	0	0	0	0	-x
9	0,16	1	1	0	0	-x
10	0,34	1	2	0	0	-x
11	0,1	0	1	0	0	-z
12	0,67	1	2	0	0	+x
13	0,34	1	4	0	0	+y
14	0,34	1	4	0	0	+y
15	0,1	0	1	0	0	-y
16	0,34	1	4	0	0	+y
17	0,34	1	3	0	0	+y
18	0,1	0	0	0	0	+y
19	0,42	1	0	0	0	+y
20	0,1	0	0	0	0	-y
21	0,85	2	2	0	0	+y
22	0,34	1	3	0	0	+z
23	0,14	0	1	0	0	+y
24	0,34	1	2	0	0	+y
25	1,5	2	5	0	0	-y

Tabella A.5: Configurazione che massimizza gli equipment rispetto a quella di partenza

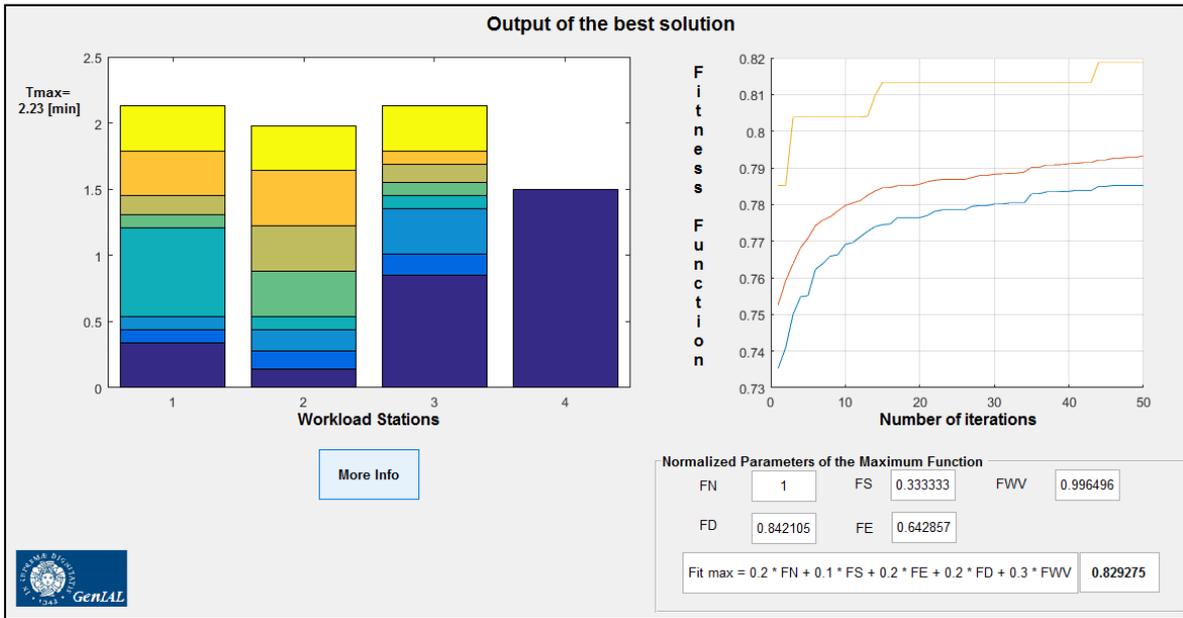


Figura A.9: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che massimizza gli equipment rispetto a quella di partenza

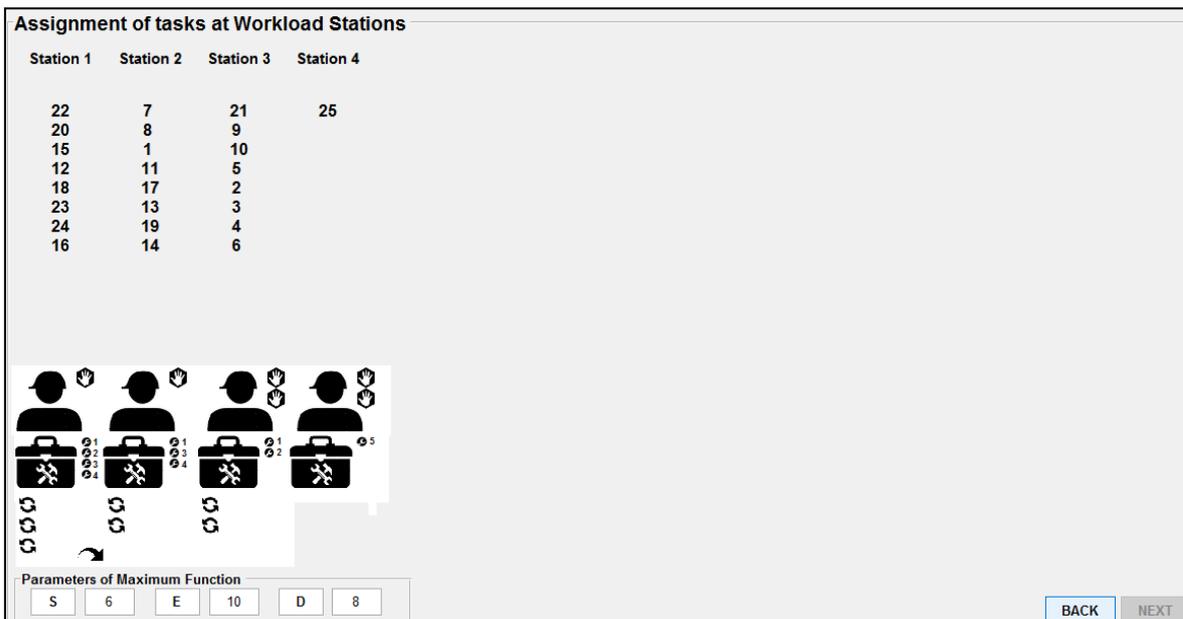


Figura A.10: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che massimizza gli equipment rispetto a quella di partenza

**Caso: Tonge**

## 1. Configurazione che minimizza le skill rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,28	0	0	0	0	-y
2	1,1	1	1	0	0	-y
3	0,9	0	0	0	0	-x
4	0,86	0	0	0	0	-y
5	0,1	0	0	0	0	+x
6	1,46	1	2	1	0	-y
7	0,35	0	0	0	0	+y
8	2,1	2	3	2	0	-y
9	1,13	1	1	0	0	-y
10	1,16	1	3	0	0	+y
11	1,4	1	0	0	0	+y
12	0,35	0	0	0	0	+z
13	2,23	1	4	3	0	-y
14	2,25	2	4	0	0	-y
15	1,56	1	0	0	0	+x
16	1,5	1	1	0	0	-y
17	0,83	0	0	0	0	-y
18	1,98	2	4	3	2	-y
19	0,31	0	0	0	0	-x
20	0,9	0	0	0	0	-x
21	0,83	2	1	0	0	-y
22	0,66	0	0	0	0	-y
23	1,21	2	3	1	0	-y
24	0,2	0	0	0	0	-z
25	2,53	2	5	1	0	-y
26	0,7	0	0	0	0	-y
27	0,75	0	4	0	0	-x
28	1,23	1	3	0	0	-x
29	0,43	0	0	0	0	-y
30	0,18	0	0	0	0	-y
31	0,51	0	1	0	0	-y
32	0,83	1	0	0	0	+y
33	1,7	0	5	2	0	-y
34	0,76	1	0	0	0	+x
35	0,58	0	6	3	0	-y
36	0,66	0	0	0	0	+x
37	0,03	0	0	0	0	+x
38	0,01	0	0	0	0	+z
39	0,05	0	0	0	0	+z
40	0,21	0	0	0	0	+z
41	0,26	0	0	0	0	+z
42	0,41	0	0	0	0	+x
43	0,35	0	3	0	0	-y
44	0,71	0	0	0	0	-y
45	0,5	0	0	0	0	-z
46	1,38	1	7	5	0	-y
47	1,48	1	8	7	0	-y
48	0,93	0	2	0	0	-y
49	0,98	1	0	0	0	+y

50	0,71	0	3	0	0	+z
51	0,18	0	0	0	0	+y
52	0,43	1	0	0	0	+y
53	0,73	0	4	0	0	-z
54	2,01	2	8	7	0	-y
55	0,63	0	0	0	0	+y
56	1,13	1	4	3	0	+x
57	0,36	0	0	0	0	+y
58	0,11	0	0	0	0	+y
59	0,26	0	0	0	0	+y
60	0,53	0	6	0	0	-x
61	0,41	0	0	0	0	+y
62	0,45	0	0	0	0	+y
63	2,6	2	8	5	1	-y
64	0,46	0	1	0	0	-y
65	0,25	0	0	0	0	+y
66	0,43	0	0	0	0	-y
67	0,3	0	0	0	0	-y
68	1,2	1	6	5	0	-y
69	0,38	0	0	0	0	-x
70	0,45	1	9	0	0	-y

Tabella A.6: Configurazione che minimizza le skill rispetto a quella di partenza

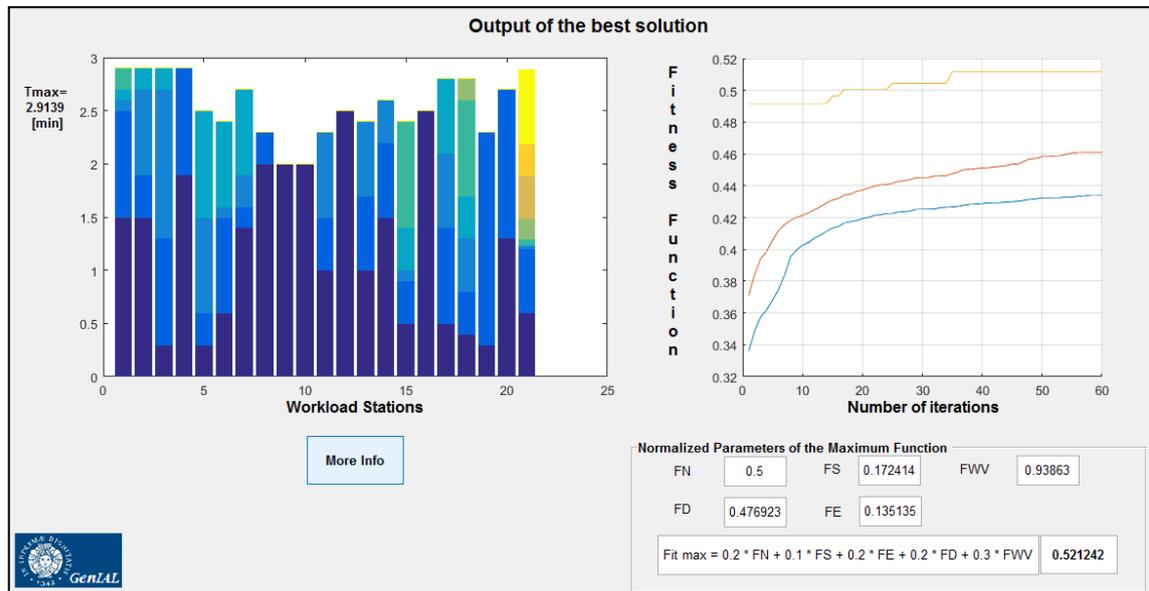


Figura A.11: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che minimizza le skill rispetto a quella di partenza

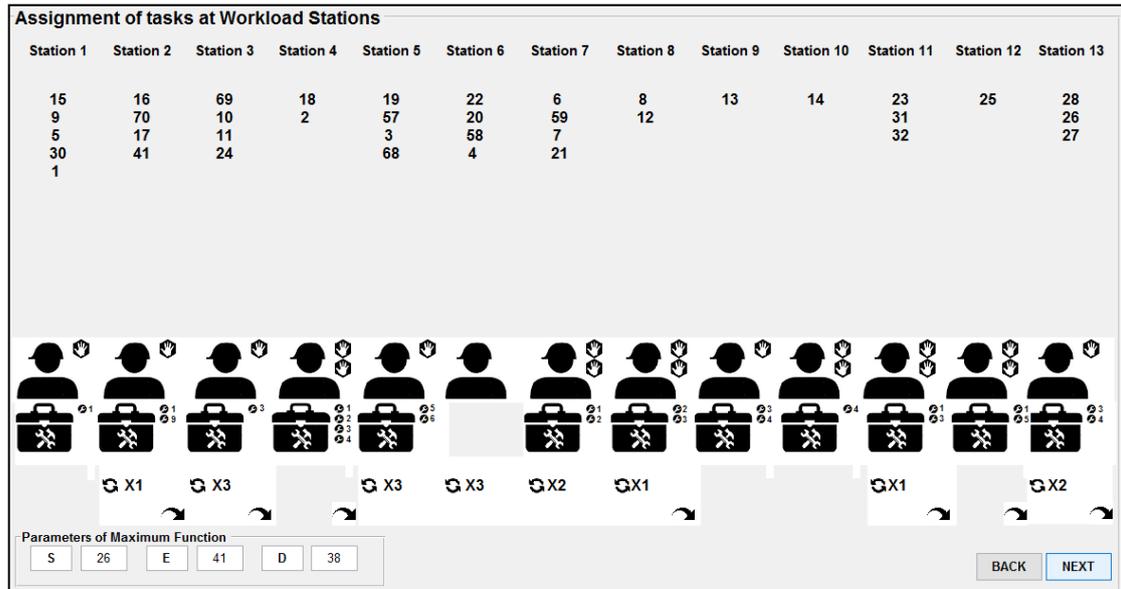


Figura A.12: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per l configurazione che minimizza le skill rispetto a quella di partenza (stazioni 1-13)

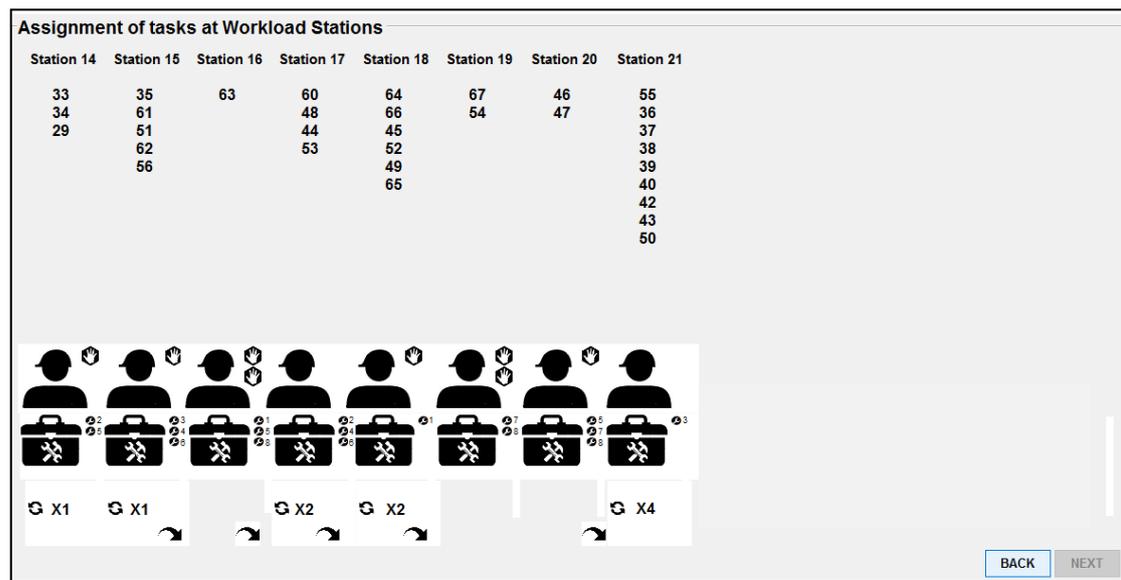


Figura A.13: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per l configurazione che minimizza le skill rispetto a quella di partenza (stazioni 14-21)

2. Configurazione che varia mediamente le skill rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,28	0	0	0	0	-y
2	1,1	2	1	0	0	-y
3	0,9	1	0	0	0	-x
4	0,86	1	1	0	0	-y
5	0,1	0	0	0	0	+x

6	1,46	1	2	1	0	-y
7	0,35	1	0	0	0	+y
8	2,1	2	3	2	0	-y
9	1,13	1	1	0	0	-y
10	1,16	2	3	0	0	+y
11	1,4	2	0	0	0	+y
12	0,35	2	0	0	0	+z
13	2,23	2	4	3	0	-y
14	2,25	2	4	0	0	-y
15	1,56	1	0	0	0	+x
16	1,5	1	1	0	0	-y
17	0,83	1	0	0	0	-y
18	1,98	2	4	3	2	-y
19	0,31	0	0	0	0	-x
20	0,9	2	0	0	0	-x
21	0,83	2	1	0	0	-y
22	0,66	2	0	0	0	-y
23	1,21	2	3	1	0	-y
24	0,2	0	0	0	0	-z
25	2,53	2	5	1	0	-y
26	0,7	1	0	0	0	-y
27	0,75	1	4	0	0	-x
28	1,23	1	3	0	0	-x
29	0,43	1	0	0	0	-y
30	0,18	0	0	0	0	-y
31	0,51	1	1	0	0	-y
32	0,83	1	0	0	0	+y
33	1,7	2	5	2	0	-y
34	0,76	1	0	0	0	+x
35	0,58	1	6	3	0	-y
36	0,66	1	0	0	0	+x
37	0,03	0	0	0	0	+x
38	0,01	0	0	0	0	+z
39	0,05	0	0	0	0	+z
40	0,21	0	0	0	0	+z
41	0,26	0	0	0	0	+z
42	0,41	1	0	0	0	+x
43	0,35	0	3	0	0	-y
44	0,71	1	0	0	0	-y
45	0,5	1	0	0	0	-z
46	1,38	2	7	5	0	-y
47	1,48	2	8	7	0	-y
48	0,93	2	2	0	0	-y
49	0,98	1	0	0	0	+y
50	0,71	1	3	0	0	+z
51	0,18	1	0	0	0	+y
52	0,43	1	0	0	0	+y
53	0,73	1	4	0	0	-z
54	2,01	2	8	7	5	-y
55	0,63	2	0	0	0	+y
56	1,13	2	4	3	0	+x
57	0,36	0	0	0	0	+y
58	0,11	0	0	0	0	+y
59	0,26	0	0	0	0	+y
60	0,53	1	6	0	0	-x
61	0,41	1	0	0	0	+y
62	0,45	1	0	0	0	+y
63	2,6	2	8	5	1	-y
64	0,46	1	1	0	0	-y
65	0,25	0	0	0	0	+y
66	0,43	1	0	0	0	-y
67	0,3	0	0	0	0	-y
68	1,2	2	6	5	0	-y

69	0,38	0	0	0	0	-x
70	0,45	1	9	0	0	-y

Tabella A.7: Configurazione che varia mediamente le skill rispetto a quella di partenza

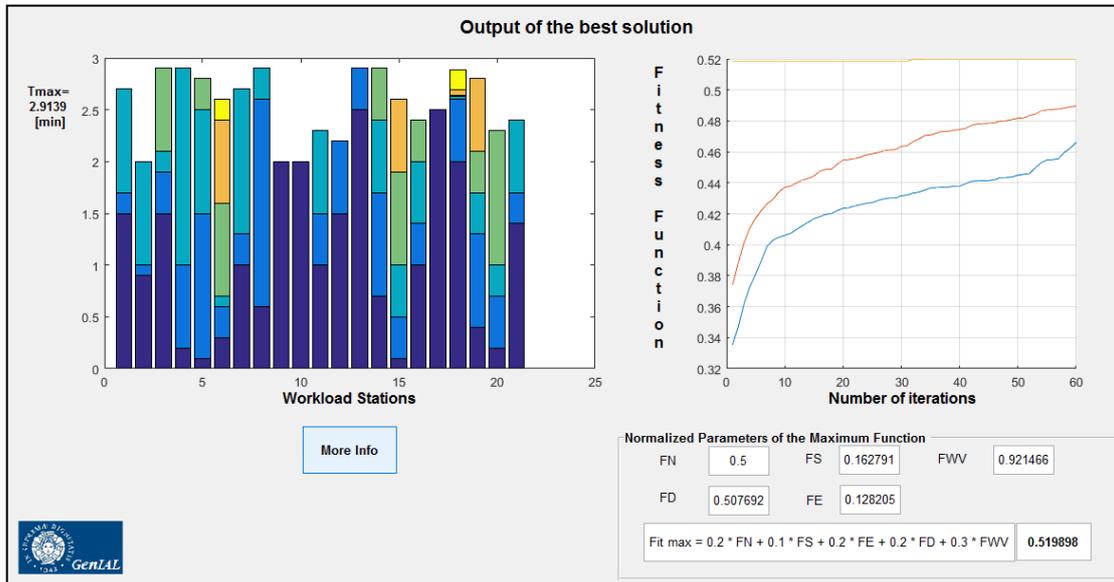


Figura A.14: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che varia mediamente le skill rispetto a quella di partenza

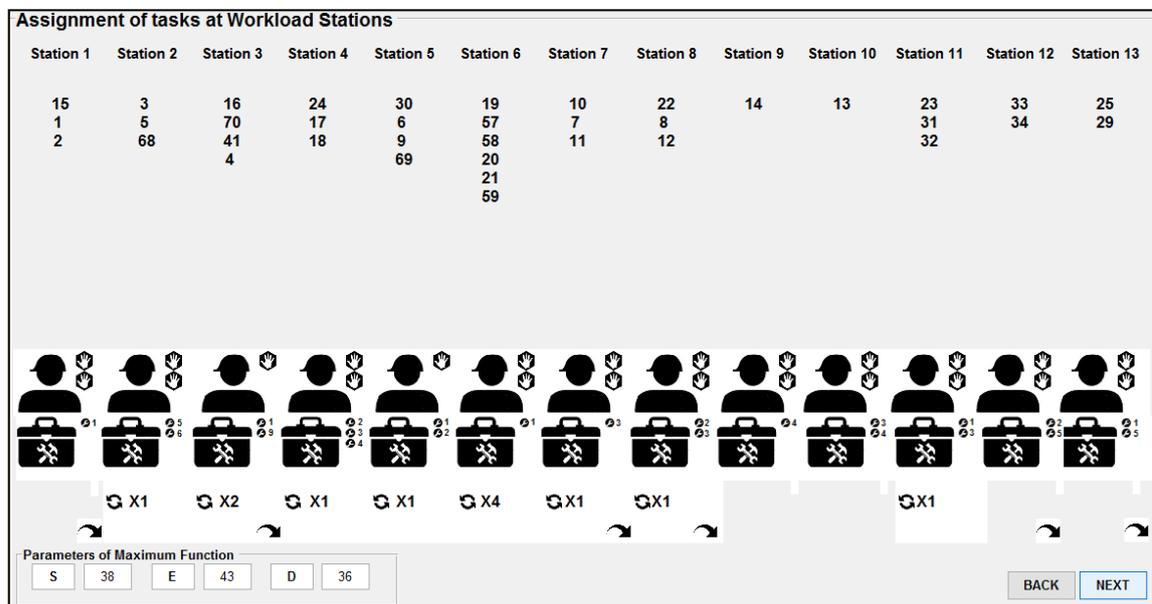


Figura A.15: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che varia mediamente le skill rispetto a quella di partenza (stazioni 1-13)

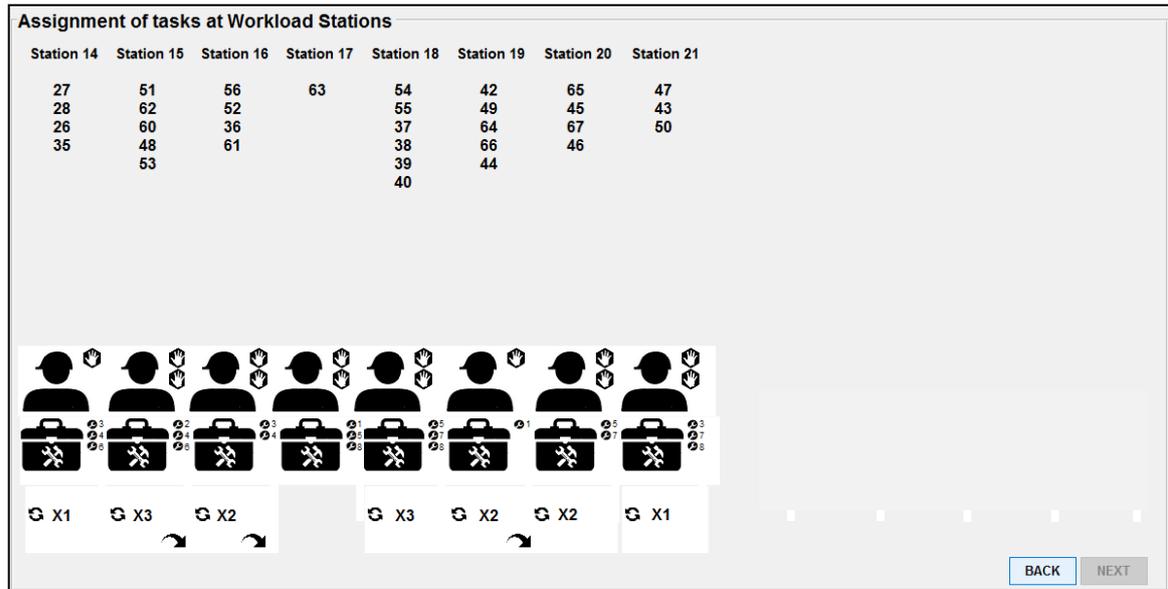


Figura A.16: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che varia mediamente le skill rispetto a quella di partenza (stazioni 14-21)

### 3. Configurazione che massimizza le skill rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,28	0	0	0	0	-y
2	1,1	1	1	0	0	-y
3	0,9	0	0	0	0	-x
4	0,86	2	1	0	0	-y
5	0,1	0	0	0	0	+x
6	1,46	2	2	1	0	-y
7	0,35	1	0	0	0	+y
8	2,1	2	3	2	0	-y
9	1,13	2	1	0	0	-y
10	1,16	2	3	0	0	+y
11	1,4	1	0	0	0	+y
12	0,35	2	0	0	0	+z
13	2,23	2	4	3	0	-y
14	2,25	2	4	0	0	-y
15	1,56	2	0	0	0	+x
16	1,5	1	1	0	0	-y
17	0,83	1	0	0	0	-y
18	1,98	2	4	3	2	-y
19	0,31	1	0	0	0	-x
20	0,9	2	0	0	0	-x
21	0,83	2	1	0	0	-y
22	0,66	1	0	0	0	-y
23	1,21	2	3	1	0	-y
24	0,2	0	0	0	0	-z
25	2,53	2	5	1	0	-y
26	0,7	1	0	0	0	-y
27	0,75	1	4	0	0	-x
28	1,23	1	3	0	0	-x
29	0,43	0	0	0	0	-y
30	0,18	0	0	0	0	-y
31	0,51	1	1	0	0	-y
32	0,83	2	0	0	0	+y

33	1,7	2	5	2	0	-y
34	0,76	1	0	0	0	+x
35	0,58	1	6	3	0	-y
36	0,66	1	0	0	0	+x
37	0,03	0	0	0	0	+x
38	0,01	0	0	0	0	+z
39	0,05	0	0	0	0	+z
40	0,21	1	0	0	0	+z
41	0,26	0	0	0	0	+z
42	0,41	0	0	0	0	+x
43	0,35	1	3	0	0	-y
44	0,71	1	0	0	0	-y
45	0,5	1	0	0	0	-z
46	1,38	2	7	5	0	-y
47	1,48	2	8	7	0	-y
48	0,93	2	2	0	0	-y
49	0,98	1	0	0	0	+y
50	0,71	1	3	0	0	+z
51	0,18	0	0	0	0	+y
52	0,43	1	0	0	0	+y
53	0,73	1	4	0	0	-z
54	2,01	2	8	7	5	-y
55	0,63	2	0	0	0	+y
56	1,13	1	4	3	0	+x
57	0,36	0	0	0	0	+y
58	0,11	0	0	0	0	+y
59	0,26	0	0	0	0	+y
60	0,53	1	6	0	0	-x
61	0,41	0	0	0	0	+y
62	0,45	1	0	0	0	+y
63	2,6	2	8	5	1	-y
64	0,46	2	1	0	0	-y
65	0,25	0	0	0	0	+y
66	0,43	2	0	0	0	-y
67	0,3	0	0	0	0	-y
68	1,2	2	6	5	0	-y
69	0,38	0	0	0	0	-x
70	0,45	1	9	0	0	-y

Tabella A.8: Configurazione che massimizza le skill rispetto a quella di partenza

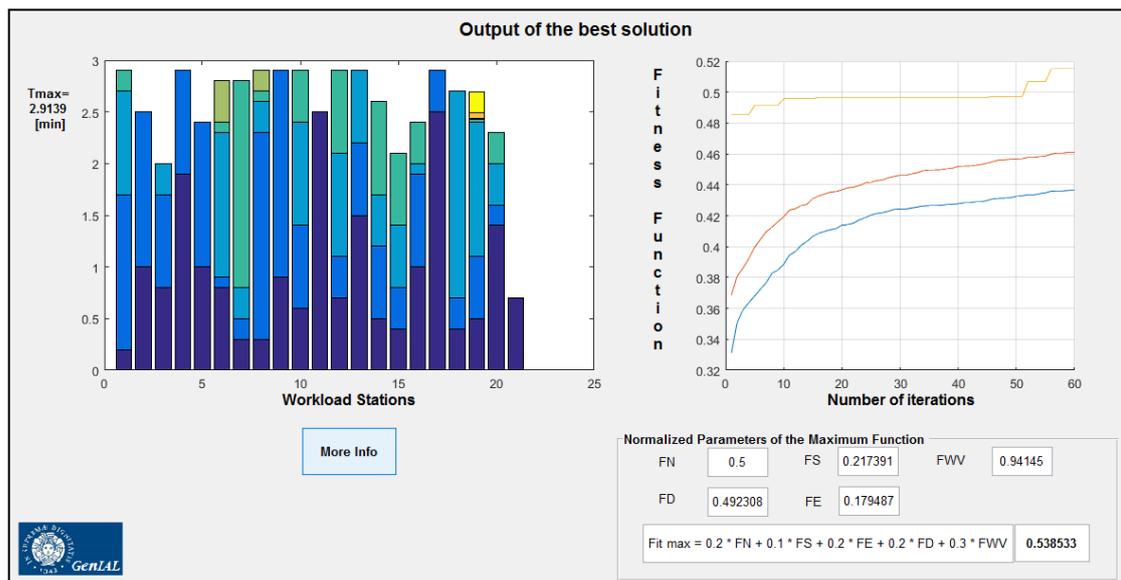


Figura A.17: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che massimizza le skill rispetto a quella di partenza

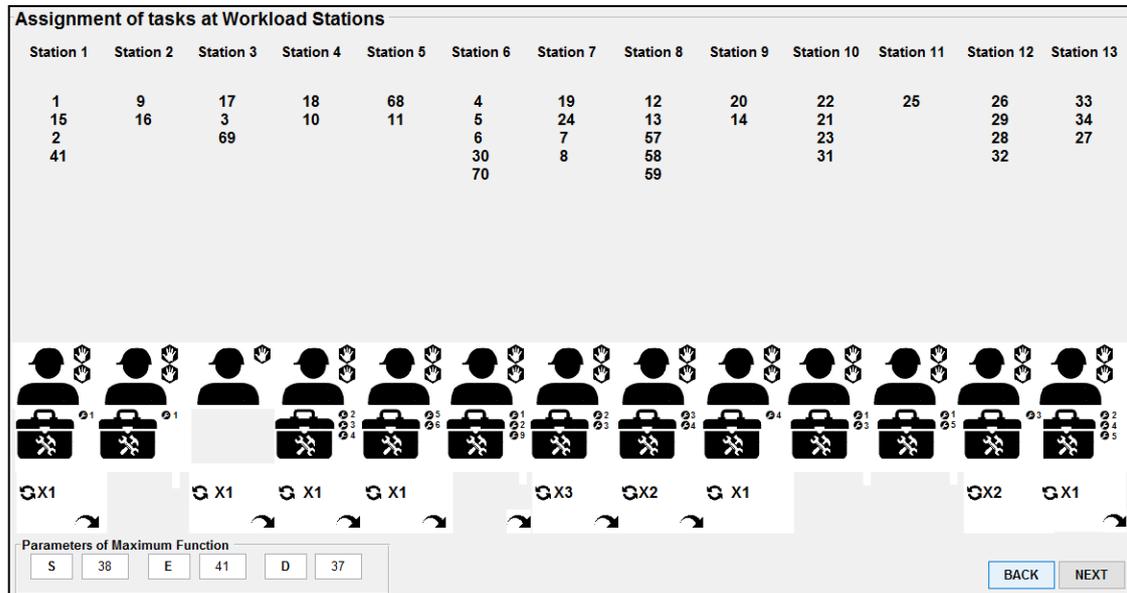


Figura A.18: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che massimizza le skill rispetto a quella di partenza (stazioni 1-13)

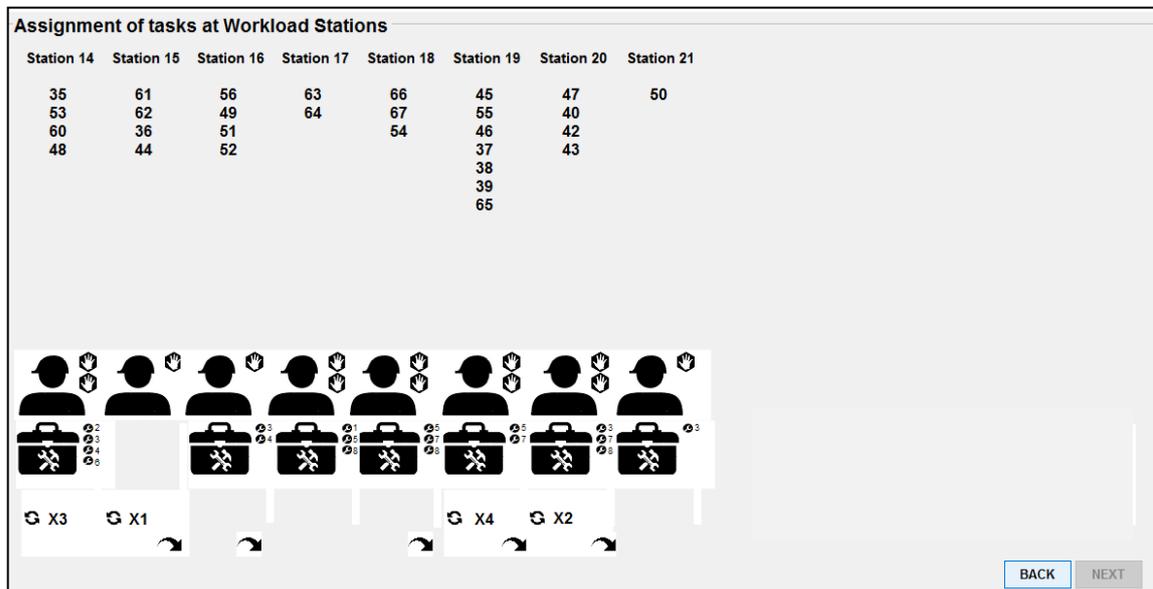


Figura A.19: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che massimizza le skill rispetto a quella di partenza (stazioni 14-21)

4. Configurazione che minimizza gli equipment rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,28	0	0	0	0	-y
2	1,1	1	0	0	0	-y
3	0,9	0	0	0	0	-x
4	0,86	0	0	0	0	-y
5	0,1	0	0	0	0	+x
6	1,46	1	1	2	0	-y
7	0,35	0	0	0	0	+y
8	2,1	2	3	2	0	-y
9	1,13	1	1	0	0	-y
10	1,16	2	3	0	0	+y
11	1,4	1	0	0	0	+y
12	0,35	2	0	0	0	+z
13	2,23	2	4	0	0	-y
14	2,25	2	3	0	0	-y
15	1,56	1	0	0	0	+x
16	1,5	1	0	0	0	-y
17	0,83	1	0	0	0	-y
18	1,98	2	4	3	2	-y
19	0,31	0	0	0	0	-x
20	0,9	2	0	0	0	-x
21	0,83	2	0	0	0	-y
22	0,66	2	0	0	0	-y
23	1,21	2	3	1	0	-y
24	0,2	1	0	0	0	-z
25	2,53	2	1	0	0	-y
26	0,7	0	0	0	0	-y
27	0,75	0	4	0	0	-x
28	1,23	1	3	0	0	-x
29	0,43	0	0	0	0	-y
30	0,18	0	0	0	0	-y
31	0,51	1	0	0	0	-y
32	0,83	1	0	0	0	+y
33	1,7	1	5	0	0	-y
34	0,76	1	0	0	0	+x
35	0,58	0	6	3	0	-y
36	0,66	1	0	0	0	+x
37	0,03	0	0	0	0	+x
38	0,01	0	0	0	0	+z
39	0,05	0	0	0	0	+z
40	0,21	0	0	0	0	+z
41	0,26	0	0	0	0	+z
42	0,41	0	0	0	0	+x
43	0,35	0	0	0	0	-y
44	0,71	1	0	0	0	-y
45	0,5	1	0	0	0	-z
46	1,38	2	7	5	0	-y
47	1,48	2	8	7	0	-y
48	0,93	2	0	0	0	-y
49	0,98	1	0	0	0	+y
50	0,71	1	3	0	0	+z
51	0,18	1	0	0	0	+y
52	0,43	1	0	0	0	+y
53	0,73	1	4	0	0	-z
54	2,01	2	8	7	0	-y
55	0,63	2	0	0	0	+y
56	1,13	1	4	0	0	+x
57	0,36	0	0	0	0	+y
58	0,11	0	0	0	0	+y
59	0,26	0	0	0	0	+y
60	0,53	1	6	0	0	-x

61	0,41	0	0	0	0	+y
62	0,45	0	0	0	0	+y
63	2,6	2	8	5	1	-y
64	0,46	2	0	0	0	-y
65	0,25	0	0	0	0	+y
66	0,43	2	0	0	0	-y
67	0,3	0	0	0	0	-y
68	1,2	1	6	0	0	-y
69	0,38	0	0	0	0	-x
70	0,45	1	9	0	0	-y

Tabella A.9: Configurazione minimizza gli equipment rispetto a quella di partenza

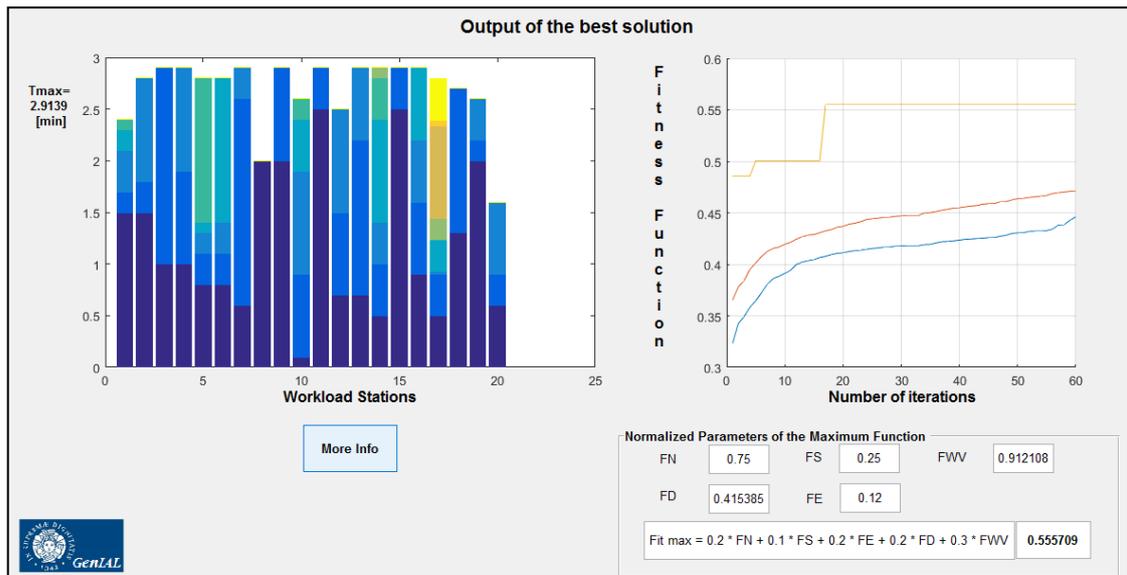


Figura A.20: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che minimizza gli equipment rispetto a quella di partenza

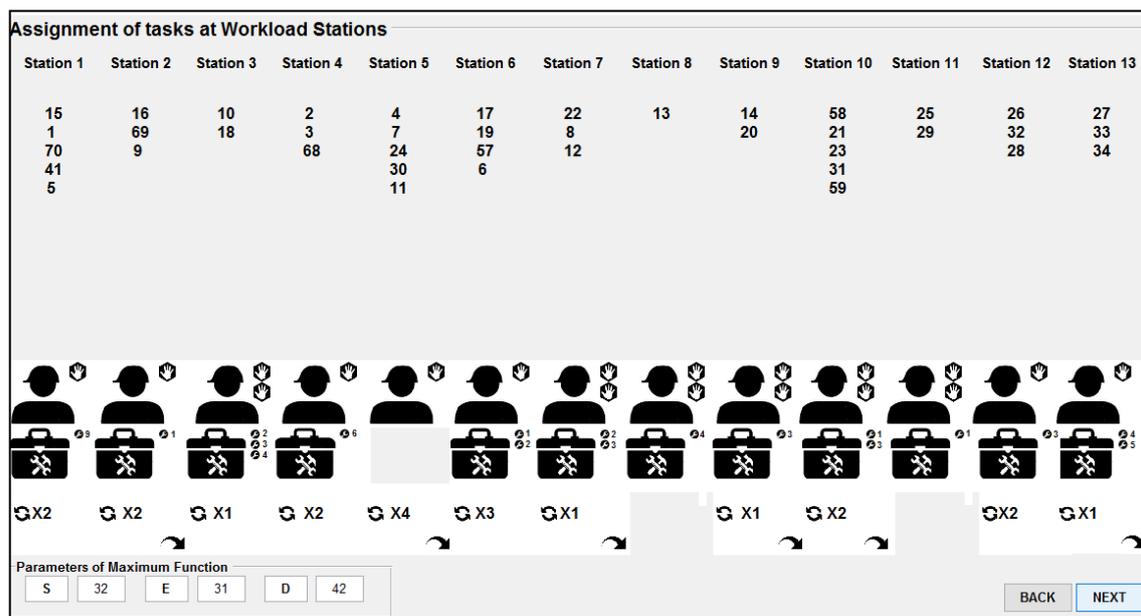


Figura A.21: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per l configurazione che minimizza gli equipment rispetto a quella di partenza (stazioni 1-13)

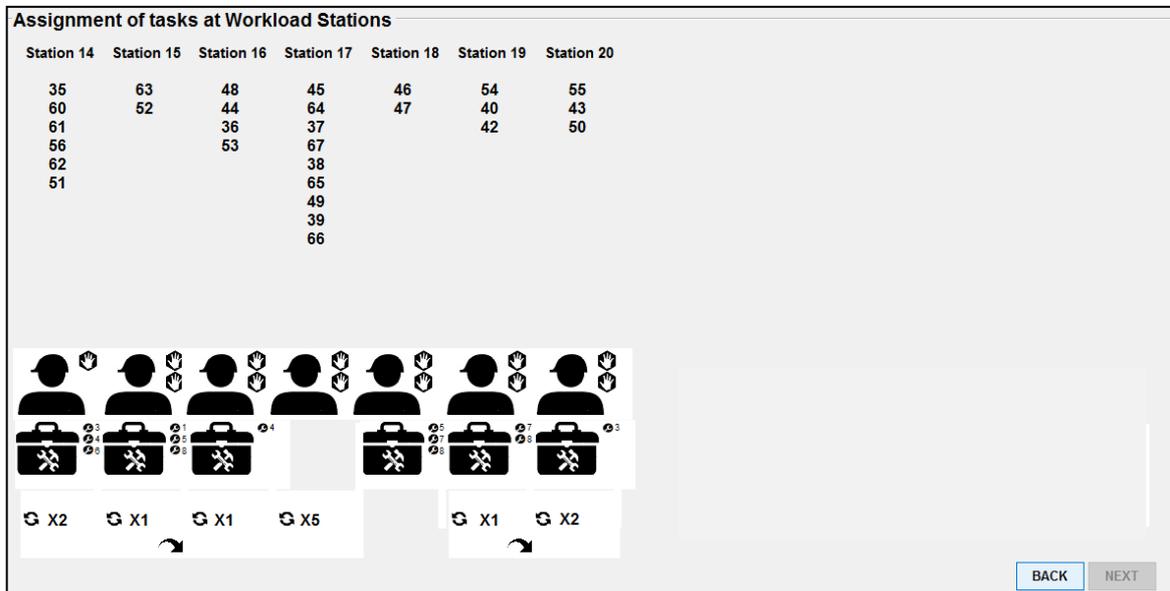


Figura A.22: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per l configurazione che minimizza gli equipment rispetto a quella di partenza (stazioni 14-20)

5. Configurazione che varia mediamente gli equipment rispetto a quella di partenza

Op	$T_{ek}$ [min]	Skill	Equip 1	Equip 2	Equip 3	Dir
1	0,28	0	0	0	0	-y
2	1,1	1	1	0	0	-y
3	0,9	0	1	0	0	-x
4	0,86	0	1	2	0	-y
5	0,1	0	0	0	0	+x
6	1,46	1	2	1	0	-y
7	0,35	0	1	0	0	+y
8	2,1	2	3	2	0	-y
9	1,13	1	1	0	0	-y
10	1,16	2	3	1	0	+y
11	1,4	1	1	0	0	+y
12	0,35	2	2	3	0	+z
13	2,23	2	4	3	0	-y
14	2,25	2	4	0	0	-y
15	1,56	1	0	0	0	+x
16	1,5	1	1	0	0	-y
17	0,83	1	0	0	0	-y
18	1,98	2	4	3	2	-y
19	0,31	0	1	0	0	-x
20	0,9	2	2	0	0	-x
21	0,83	2	1	0	0	-y
22	0,66	2	0	0	0	-y
23	1,21	2	3	1	0	-y
24	0,2	1	0	0	0	-z
25	2,53	2	5	1	2	-y
26	0,7	0	1	0	0	-y
27	0,75	0	4	0	0	-x

28	1,23	1	3	0	0	-x
29	0,43	0	3	0	0	-y
30	0,18	0	0	0	0	-y
31	0,51	1	1	0	0	-y
32	0,83	1	0	0	0	+y
33	1,7	1	5	2	0	-y
34	0,76	1	1	0	0	+x
35	0,58	0	6	3	0	-y
36	0,66	1	0	0	0	+x
37	0,03	0	0	0	0	+x
38	0,01	0	0	0	0	+z
39	0,05	0	0	0	0	+z
40	0,21	0	1	0	0	+z
41	0,26	0	0	0	0	+z
42	0,41	0	2	0	0	+x
43	0,35	0	3	0	0	-y
44	0,71	1	0	0	0	-y
45	0,5	1	4	1	0	-z
46	1,38	2	7	5	0	-y
47	1,48	2	8	7	0	-y
48	0,93	2	2	0	0	-y
49	0,98	1	0	0	0	+y
50	0,71	1	3	0	0	+z
51	0,18	1	0	0	0	+y
52	0,43	1	0	0	0	+y
53	0,73	1	4	0	0	-z
54	2,01	2	8	7	5	-y
55	0,63	2	0	0	0	+y
56	1,13	1	4	3	0	+x
57	0,36	0	0	0	0	+y
58	0,11	0	0	0	0	+y
59	0,26	0	2	0	0	+y
60	0,53	1	6	0	0	-x
61	0,41	0	0	0	0	+y
62	0,45	0	1	0	0	+y
63	2,6	2	8	5	1	-y
64	0,46	2	1	0	0	-y
65	0,25	0	0	0	0	+y
66	0,43	2	1	0	0	-y
67	0,3	0	2	0	0	-y
68	1,2	1	6	5	0	-y
69	0,38	0	0	0	0	-x
70	0,45	1	9	0	0	-y

Tabella A.10: Configurazione varia mediamente gli equipment rispetto a quella di partenza

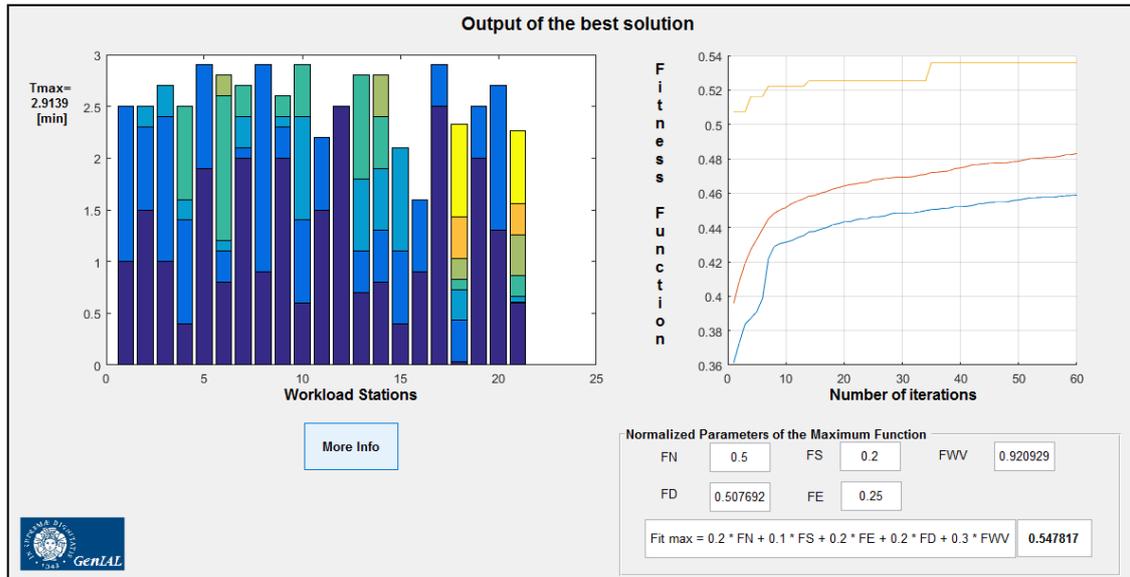


Figura A.23: Interfaccia grafica per la visualizzazione del miglior output per la configurazione che varia mediamente gli equipment rispetto a quella di partenza

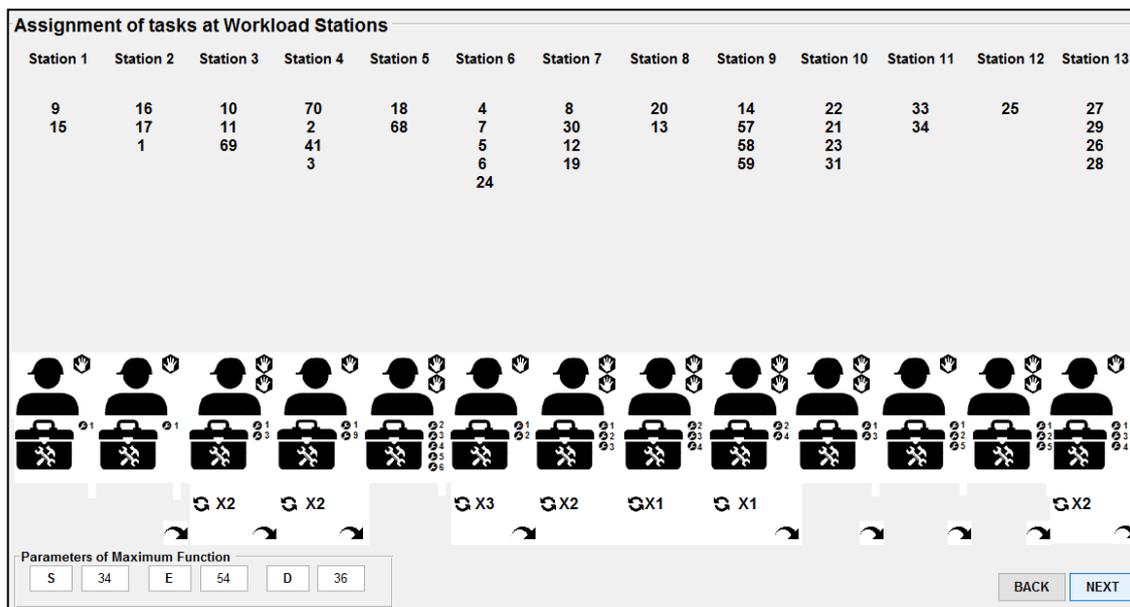


Figura A.24: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che varia mediamente gli equipment rispetto a quella di partenza (stazioni 1-13)

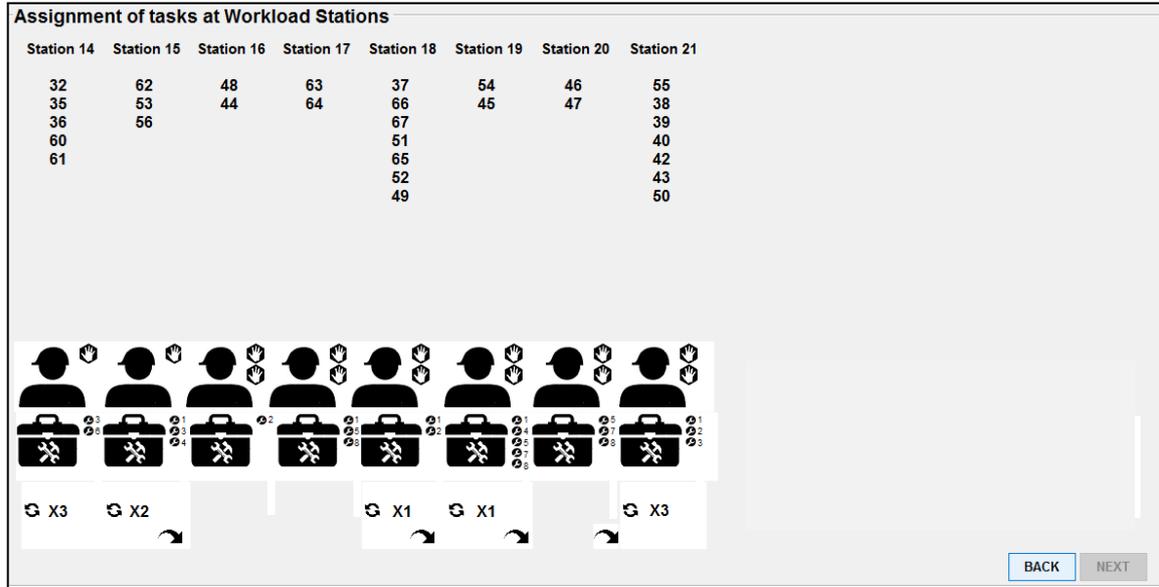


Figura A.25: Interfaccia grafica che dettaglia le caratteristiche di ciascuna stazione per la configurazione che varia mediamente gli equipment rispetto a quella di partenza (stazioni 14-21)

## Bibliografia

1. **Dini, Gino.** Sistemi Integrati di Produzione. 2013.
2. **Scholl, Armin.** *Balancing and Sequencing of Assembly Lines.*
3. **Tarek Al-Hawari, Marwan Ali, Omar Al-Araidah, Ahmad Mumani.** Development of a genetic algorithm for multi-objective assembly line balancing using multiple assignment approach. 11 November 2014.
4. **Seren Oz Mehmet Tasan, Semra Tunali.** A review of the current applications of genetic algorithms in assembly line balancing. 2007.
5. **Paganella, Alessandra.** *Sviluppo e applicazione di uno strumento software a supporto del bilanciamento di linee di montaggio manuali.* 2013.
6. Wikipedia. [Online]
7. **Alexandre Dolgui, Jean-Marie Proth.** *Supply Chain Engineering.* 2010.
8. **J. Rubinovitz, G. Levitin.** Genetic algorithm for assembly line balancing.
9. **JOSE FERNANDO GONCALVES, JORGE RAIMUNDO DE ALMEIDA.** A Hybrid Genetic Algorithm for Assembly Line Balancing. 2002.
10. **Brudaru O., Valmar B.** Genetic algorithm with embryonic chromosomes for assembly line balancing with fuzzy processing times. 2004.
11. **E., Falkenauer.** A genetic algorithm for grouping. 1991.
12. **Stephen Chen, Stephen F. Smith.** Commonality and Genetic Algorithms. 1996.
13. **J. Holland.** *Adaptation in natural and artificial system.* 1975.
14. **E. Falkenauer, A. Delchambre.** A Genetic Algorithm for Bin Packing and Line Balancing. 1992.
15. **Leu Y. Y., Matheson, L.A., Rees, L.P.** Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Population and Multiple Evaluation Criteria. 1994.
16. **Yasuhiro TSUJIMURA, Mitsuo GEN, Erika KUBOTA.** Solving fuzzy Assembly-line Balancing Problem with Genetic Algorithms. 1995.
17. **Falkenauer, E.** A Hybrid Grouping Genetic Algorithm. 1996.
18. **YEO KEUN KIM, YONG JU KIM, YEONGHO KIM.** GENETIC ALGORITHMS FOR ASSEMBLY LINE BALANCING WITH VARIOUS OBJECTIVES.
19. **Mitsuo Gen, Yasuhiro Tsujimura, Yinxiu Li.** Fuzzy Assembly line balancing using genetic algorithms. 1996.

20. **TADAHIKO MURATA, HISAO ISHIBUCHI and HIDEO TANAKA.** MULTI-OBJECTIVE GENETIC ALGORITHM AND ITS APPLICATIONS TO FLOWSHOP SCHEDULING. 1996.
21. **Falkenauer, E.** A grouping genetic algorithm for line balancing with resource dependent task time. 1997.
22. *A Multi-Objective Genetic Local Search Algorithm.* **Hisao Ishibuchi, Tadahiko Murata.** 1998.
23. **Yong Ju Kim, Yeo Keun Kim and Yongkyun Cho.** A HEURISTIC-BASED GENETIC ALGORITHM FOR WORKLOAD SMOOTHING IN ASSEMBLY LINES. 1998.
24. **Sabuncuoglu I, . Erel E., Tanyer M.** Assembly line balancing using genetic algorithms. 2000.
25. **S.G. Ponnambalam, P. Aravindan, G. Mogileeswar Naidu.** A Multi-Objective Genetic Algorithm for Solving Assembly Line Balancing Problems.
26. **Joaquin Bautista, Raul Suarez, Manuel Mateo, Ramon Companys.** Local Search Heuristics for the Assembly Line Balancing Problem with Incompatibilities Between Tasks. 2000.
27. **Ruey-Shun Chen, Kun-Yung Lu, Shien-Chiang Yu.** A hybrid genetic algorithm approach on multi-objective of assembly planning problem. 2002.
28. **Evelyn C. Brown, Robert T. Sumichrast.** Evaluating performance advantages of grouping genetic algorithms. 2004.
29. *Use of genetic algorithms in operations management Part 1: Applications.* **D J Stockton, L Quinn, R A Khalil.** 2004.
30. **Gen M, Cheng R, Lin L.** *Network models and optimization: multiobjective genetic algorithm approach.* 2008.
31. **Yin, Jianfeng Yu & Yuehong.** Assembly line balancing based on an adaptive genetic algorithm. 2009.
32. **Qihua Tang, Zhonghua Xiao, Yanli Liang, Mingxing Deng, Zhongmin Xi.** Novel Approach for Balancing Manual Automobile Assembly Based on Genetic Algorithm. 2010.
33. **Octav Brudaru, Cristian Rotaru.** Dynamic segregative genetic algorithm for assembly lines balancing. 2010.
34. **Noraini Mohd Razali, John Geraghty.** Biologically Inspired Genetic Algorithm to Minimize Idle Time of the Assembly Line Balancing. 2011.
35. **Jiahua Weng, Xianchao Wu, Hisashi Onari.** A MULTI-OBJECTIVE APPROACH BASED ON GENETIC ALGORITHM FOR MULTI-MODEL LINE PROCESS PLANNING CONSIDERING DIFFERENCE IN WORKER ABILITY. 2011.
36. **Su, Jianping Dou & Jun Li & Chun.** A novel feasible task sequence-oriented discrete particle swarm algorithm for simple assembly line balancing problem of type 1. 2013.

37. **Giandomenici, Giulia.** Sviluppo e Applicazione di una Metodologia per il Miglioramento dell'Efficienza Produttiva nel Settore Pelletteria. 2013.
38. **Kilinci, Ozcan.** Firing sequences backward algorithm for simple assembly line balancing problem of type 1. 2008.
39. **Tonge, Fred. M.** A HEURISTIC PROGRAM FOR ASSEMBLY LINE BALANCING. 1960.
40. **Hoffmann, Thomas R.** EUREKA: A HYBRID SYSTEM FOR ASSEMBLY LINE BALANCING. 1992.
41. **Wenqiang Zhang, Mitsuo Gen and Lin Lin.** A Multiobjective Genetic Algorithm for Assembly Line Balancing Problem with Worker Allocation. 2008.
42. **Shimon Y. Nof, Wilbert E. Wilhelm, Hans-Jurgen Warnecke.** *Industrial assembly.* 1997.
43. **Fan Shu, Zi-Qi-Zu, Chao Mi e Xiao-Ming Yang.** Research on Engine Assembly Line Balancing Based on an Improved Genetic Algorithm. 2013.
44. **Edward J. Anderson, Michael C. Ferris.** Genetic Algorithms for combinatorial Optimization: The Assembly Line Balancing Problem. January 1993.
45. **Armin Scholl, Christian Becker.** State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. 11 September 2004.