

UNIVERSITÀ DI PISA



FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E  
NATURALI

CORSO DI LAUREA IN INFORMATICA

**Realizzazione di un software di  
simulazione della catena di  
processi produttivi di  
un'azienda siderurgica**

*Tutore Accademico:*

Prof. Fabio Gadducci

*Tutore Aziendale:*

Ing. Valentina Colla

*Candidato:*

Marco Del Seppia

27 Giugno, 2014



a Nonna Piera,

la tua infinita pazienza mi ha aiutato a sopportare i momenti difficili.

a Babbo e Mamma,

avete avuto un bel coraggio.

# Sommario

La pianificazione della produzione dell'acciaio incentrata su un modello basato sulla qualità, è un tema che si presta a continue innovazioni nell'ambito della ricerca sulle leghe di ferro e carbonio. I sistemi disponibili sulla produzione programmata non tengono di parametri qualitativi nelle loro funzioni obiettivo e algoritmi, infatti esiste solo una buona base teorica pubblicata in letteratura, per le ottimizzazioni multi obiettivo [8].

La ricerca nell'acciaio è ancora focalizzata sull'ottimizzazione di parametri come la forza di laminazione, la coppia, la tensione e la potenza, i quali sono necessari per garantire l'efficienza produttiva e ridurre i costi, tuttavia, solo poche prodotti prestano attenzione alla programmazione e pianificazione della produzione.

Di fronte alle più severe esigenze dei clienti, in termini di diversità dei prodotti, di capacità di consegna e costante qualità del prodotto, la pianificazione della produzione nell'area di laminazione a freddo, è diventata un importante problema che collega la gestione della produzione con la laminazione stessa. In particolare, i produttori di acciaio aventi più di una catena di produzione devono destinare la produzione a vari impianti, a seconda delle loro capacità e la possibilità di produrre strisce di alta qualità.

Questo progetto mira a implementare un sistema di assistenza technology-based per la programmazione della produzione e pianificazione. Technology-based significa che i modelli di input-output sono costruiti per prevedere il tempo di lavorazione e la qualità finale dell'acciaio, in termini di flatness (planarità), crossbow (distorsione a balestra), process workload (carico di lavoro del processo) come parametri di qualità. I modelli di previsione identificati e gli indici di qualità definiti, sono stati poi inclusi nel compito di ottimizzare la pianificazione della produzione per la catena di processi nella cold rolling area (area di laminazione a freddo): cold rolling (laminazione a freddo, quattro linee) → annealing (ricottura, tre linee) → temper mill (treno di rinfrescamento, tre linee) → finishing line (linea finale, numero variabile). Questo approccio, completamente nuovo, non è incluso nella produzione tradizionale dei sistemi disponibili sul mercato.

# Abstract

The topic of quality-model-based production planning has not been treated in the Steel Research at all. Available production planning systems do not consider quality parameters in their objective functions and algorithms. There is only a good theoretical basis for multiobjective optimisation, published in the literature [8].

Therefore, the challenging goal of this proposal to develop an overall quality-model-based assistance system can be considered as a breakthrough in the steel production planning technology. Steel research is still focused on optimising parameters such as rolling force, torque, tension and power, which are necessary to ensure production efficiency and reduce costs. However, only a few works paid attention to production scheduling and planning. Facing more stringent customer requirements in terms of product diversity, delivery capability, on-time delivery and consistent product quality, production planning in cold rolling has become a significant problem that connects the production management with the rolling process itself. In particular, steel producers having more than one production route must allocate sheet production to the various plants, depending on their capacities and capability to produce high-quality strips.

This proposal aims to strengthen the European stainless steel industry by developing a technology-based assistance system for production scheduling and planning. “Technologybased” means that input-output models are derived to predict transfer/final quality in terms of flatness, cross-bow, thickness tolerance, elongation and other quality parameter. The prediction models identified and suitable quality indices to be defined are then included in the optimisation task of production planning for the production chain: cold rolling (four plants) → annealing (three plants) → temper rolling (three plants). This approach is completely new, it is not included in traditional production systems available at the market.

# Indice

<b>Elenco delle figure</b>	<b>vii</b>
<b>Elenco delle tabelle</b>	<b>ix</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 L'azienda . . . . .	2
1.2 Panoramica del progetto . . . . .	3
1.3 Struttura della tesi . . . . .	4
<b>2 L'impianto</b>	<b>6</b>
2.1 Area di laminazione a freddo . . . . .	6
2.1.1 Cold Rolling . . . . .	8
2.1.2 Annealing . . . . .	8
2.1.3 Temper Mill . . . . .	9
2.1.4 Finishing Line . . . . .	10
2.2 Descrizione del problema . . . . .	11
2.2.1 Workload . . . . .	14
2.2.2 Crossbow . . . . .	15
2.2.3 Flatness . . . . .	15
2.3 Inclusione delle richieste sulla qualità nei requisiti per la pianificazione . . . . .	16
2.4 Planning della produzione basato su un modello generale . . . . .	18
<b>3 Problemi multi-obiettivo</b>	<b>21</b>
3.1 Algoritmi evolutivi multi-obiettivo . . . . .	22
3.2 Algoritmi Genetici . . . . .	25
3.3 Codifica del cromosoma . . . . .	27
3.4 Funzione di Creazione . . . . .	28
3.5 Funzione di Crossover . . . . .	29
3.6 Funzione di Mutazione . . . . .	30
3.7 Funzione di Fitness . . . . .	31

<b>4 Sviluppo Libreria</b>	<b>32</b>
4.1 Ambiente di sviluppo . . . . .	32
4.2 Pattern architetturale . . . . .	33
4.2.1 Modelli . . . . .	37
4.3 Schema generale . . . . .	37
4.3.1 Controller . . . . .	39
4.3.2 View . . . . .	42
4.4 Design Pattern Strategy . . . . .	42
4.5 Implementazione funzione di fitness . . . . .	43
4.6 Ottimizzazione . . . . .	45
4.7 Un possibile scenario . . . . .	52
<b>5 Interfaccia grafica</b>	<b>54</b>
5.1 Layout . . . . .	55
5.2 Modulo di input . . . . .	57
5.3 Area fasi di produzione . . . . .	59
5.4 Pianificazione dell'itinerario . . . . .	60
5.5 Grafico fronte di pareto . . . . .	61
5.6 Modulo di salvataggio dell'output . . . . .	62
5.7 Flusso di controllo . . . . .	63
<b>6 Risultati</b>	<b>65</b>
6.1 Creazione del test-set . . . . .	66
6.2 Valutazione dei risultati . . . . .	68
<b>7 Conclusioni e prospettive future</b>	<b>71</b>
<b>A Acronimi</b>	<b>73</b>
<b>Bibliografia</b>	<b>76</b>
<b>Ringraziamenti</b>	<b>77</b>

# Elenco delle figure

2.1	Diagramma globale dell'impianto . . . . .	7
2.2	Schema del processo di laminazione . . . . .	8
2.3	Processo di annealing, due blocchi da lavorare vengono riscaldati	9
2.4	Output di una linea di temper mill . . . . .	10
2.5	Percorso di produzione dipendente dalle previsioni sulla qualità e sulle esigenze di qualità del cliente . . . . .	11
2.6	Ottimizzazione multi-criterio dei percorsi di produzione, prendendo in considerazione la qualità del processo e l'output di produzione . . . . .	12
2.7	Esempio di rotolo affetto da crossbow . . . . .	15
2.8	Rappresentazione di un pezzo non perfettamente spianato . .	16
2.9	Processo di simulazione del percorso . . . . .	17
2.10	Diagramma dell'area di laminazione a freddo . . . . .	18
2.11	Possibile accoppiamento tra rotoli e ordini al termine della catena di processi . . . . .	19
3.1	Esempi di dominanza (a sinistra) e del Fronte di Pareto (a destra) in una MOP di minimizzazione con due funzioni obiettivo	23
3.2	Diagramma di flusso di un generico algoritmo genetico . . .	26
3.3	Codifica del cromosoma scelta, identificazione del macrogene, del gene, e del significato assunto da ognuno di essi . . . . .	28
4.1	Pattern MVC puro . . . . .	35
4.2	Variante pattern MVC . . . . .	36
4.3	Schema delle classi della libreria . . . . .	38
4.4	Esempio di struttura dello strategy pattern . . . . .	42
4.5	Esempio di assegnamento tra rotoli e ordini in un MBMP . .	47
4.6	Contenuto del file che racchiude i valori dei KQI generati casualmente per ogni possibile rottura di un rotolo . . . . .	49
4.7	Diagramma di sequenza . . . . .	53
5.1	Schermata principale prima dell'esecuzione della simulazione .	55

5.2	Schermata principale dopo l'esecuzione della simulazione . . . . .	56
5.3	Modulo di input compilato . . . . .	57
5.4	Contenuto dei file relativi alla lista delle bobine da lavorare (a sinistra) e agli ordini (a destra) . . . . .	58
5.5	Esempio di file di configurazione . . . . .	59
5.6	Possibile configurazione dell'impianto . . . . .	59
5.7	Esempio di pianificazione fornita dal GA . . . . .	60
5.8	Lista degli accoppiamenti tra rotoli e ordini . . . . .	61
5.9	Esempi di grafico del fronte di pareto . . . . .	62
5.10	Modulo di salvataggio dell'output . . . . .	63
5.11	Diagramma di flusso relativo alla GUI . . . . .	64

# Elenco delle tabelle

2.1	Traduzione dei target in funzioni obiettivo . . . . .	14
6.1	File relativo agli ordini usato per il testing . . . . .	67
6.2	File relativo ai rotoli disponibili usato per il testing . . . . .	67
6.3	Dati relativi al primo test, 3 ordini e 3 rotoli . . . . .	68
6.4	Dati relativi al secondo test, 9 rotoli e 9 ordini . . . . .	69
6.5	Dati relativi al secondo test, 9 rotoli e 9 ordini . . . . .	70

# Capitolo 1

## Introduzione

I sistemi di pianificazione della produzione sono essenziali per minimizzare i costi e massimizzare la produttività negli impianti di lavorazione dell'acciaio.

Questo progetto di tirocinio mira a implementare un nuovo sistema di assistenza che determina il percorso ottimale di produzione per l'acciaio inossidabile, in base ai valori di più indici di performance, grazie ai quali è possibile valutare qualità, funzionalità desiderate ed esigenze dei clienti. Il percorso viene determinato all'interno dell'aera di laminazione a freddo che comprende una serie di sottoprocessi che hanno lo scopo di conferire determinate caratteristiche meccaniche, di qualità e di finitura al materiale e di trasformare il prodotto semilavorato in prodotto finito. La laminazione a freddo è molto importante poiché interviene sulla struttura dei grani cristallini dell'acciaio e conferisce una migliore finitura superficiale (poiché il pezzo, contrariamente alla laminazione a caldo, non si ossida). Inoltre, questo processo, può servire a conferire al rotolo alcune caratteristiche fisiche desiderate, come lo spessore.

L'obiettivo principale è stato quello di sviluppare una soluzione flessibile e modulare, che sarà impiegata per ottimizzare la pianificazione dei diversi processi nell'area di laminazione a freddo, con il vantaggio che questa soluzione potrebbe essere adattata e trasferita ad altri impianti o scenari industriali.

In particolare, il tirocinio si è concentrato sulla realizzazione dell'astrazione dell'impianto contenente l'area di laminazione a freddo, della libreria

che racchiude la logica del simulatore, l'implementazione di un algoritmo di ottimizzazione e infine della realizzazione di un'interfaccia grafica tramite cui poter utilizzare la libreria sviluppata.

## 1.1 L'azienda

Il lavoro di tirocinio è stato svolto presso la divisione SIAD (Steel and Industrial Automation Division) del laboratorio PERC.RO. (PERCeptual RObotics), uno dei laboratori di ingegneria della Scuola Superiore Sant'Anna di Studi Universitari e perfezionamento. L'attività di ricerca di questo gruppo si dedica all'automazione industriale, con particolare attenzione per l'industria siderurgica. Nello specifico vengono effettuate ricerche su automazione dei processi e delle tecniche di controllo di processo, per l'applicazione di statistiche e tecniche di intelligenza artificiale per l'analisi dei dati, simulazione di processo, monitoraggio e controllo nella prospettiva dello sviluppo di sistemi intelligent manufacturing<sup>1</sup>. Il gruppo sta inoltre sviluppando attività di ricerca nel campo della modellazione matematica dei processi industriali.

L'esperienza maturata all'interno del laboratorio è risultata formativa sotto molto aspetti: dall'introduzione a problemi di ottimizzazione multiobiettivo tramite l'utilizzo di algoritmi genetici (questa parte verrà approfondita nei capitoli successivi) fino ad arrivare a sviluppare vari moduli software in linguaggi di programmazione su cui non è mai stato possibile lavorare in ambito accademico. In particolare lo sguardo è stato rivolto a linguaggi quali C++ e C# utilizzati appunto per lo sviluppo dell'intero progetto. Sia il tutor aziendale da cui sono stato affiancato, sia i collaboratori esterni al progetto, si sono sempre dimostrati disponibili a dare una mano nelle situazioni di difficoltà, durante lo sviluppo del software.

Fin dall'inizio, sono stato coinvolto nella vita quotidiana dell'ambiente lavorativo e questo mi ha permesso di sviluppare un più che soddisfacente networking con i collaboratori dell'azienda.

---

<sup>1</sup>L'utilizzo della tecnologia all'interno del processo produttivo in grado di adattarsi automaticamente ai cambiamenti ambientali e diversi requisiti di processo, con la capacità di produzione di vari prodotti con controllo e assistenza minimi da parte degli operatori

Verso la fine del tirocinio è stato richiesta la stesura di una documentazione in lingua inglese, che è stata utilizzata come report per l'ente coordinatore del progetto, il BFI (Betriebsforschungsinstitut), e mi è stato permesso di poter introdurre la relazione stessa ai responsabili tedeschi. Quest'ultima esperienza mi ha permesso un grado di coinvolgimento e di soddisfazione inaspettato.

## 1.2 Panoramica del progetto

Il software sviluppato fa parte di uno dei work package presenti all'interno del progetto di ricerca europeo RFCS (Research Fund for Coal and Steel, Fondo di ricerca carbone e acciaio) denominato TECPLAN (TEChnology-based assistance system for production PLANning in stainless steel mills, Sistema di assistenza basato sulla tecnologia per la pianificazione della produzione negli impianti di produzione di acciaio inox). Questo progetto è il primo che mira a sviluppare e testare industrialmente metodi di pianificazione incentrati su modelli della produzione, basati su indici sulla qualità nelle catene di lavorazione dell'acciaio.

Più in dettaglio, il progetto comprende lo sviluppo, l'implementazione e il testing delle seguenti caratteristiche innovative nell'automazione della lavorazione dell'acciaio:

1. Sviluppo del concetto di feedback sulla qualità nella pianificazione della produzione, dove considerare esplicitamente i principali parametri qualitativi in termini di flatness (planarità), crossbow<sup>2</sup> (balestra), thickness tolerance (tolleranza dello spessore), ecc nella pianificazione della produzione. Al contrario, i sistemi di pianificazione della produzione di oggi non fanno prendono in considerazione la qualità del prodotto finale;
2. Sviluppo di modelli basati sulla qualità per la previsione del tempo di lavorazione per ogni fase di produzione, incentrati su dati rilevati

---

<sup>2</sup>Nel campo siderurgico il termine crossbow viene utilizzato per denotare un tipo di difetto di lavorazione molto simile a quello di flatness

dai sistemi di misura dei parametri sopra menzionati, al fine di fornire un indice di applicabilità in funzione della capacità del mulino, il throughput e le esigenze del cliente;

3. Progettazione di un sistema di assistenza con un modello generale basato sulla qualità che determina automaticamente il percorso di produzione ottimale all'interno di una catena di produzione multi-percorso (laminazione a freddo → ricottura → treno di rinvenimento → linea di finitura) per l'acciaio inossidabile. Inoltre, una nuova generazione di sistemi di pianificazione della produzione automatica che va oltre i sistemi odierni, che spesso ancora si basano sull'esperienza e il feeling del progettista;
4. Sviluppo di algoritmi di ottimizzazione multi-obiettivo adatti a risolvere il complesso problema di pianificazione della produzione, per trovare il miglior compromesso tra i vari aspetti relativi alla qualità e l'utilizzo il più omogeneo possibile di tutte le fasi di lavorazione lungo tutta catena. Saranno utilizzati particolari paradigmi di algoritmi genetici in quanto possono risolvere problemi arbitrariamente complessi, problemi non lineari con vincoli complessi garantendo la fattibilità delle soluzioni.

Si può notare che in effetti, alcuni altri progetti (non menzionati qui) hanno affrontato il problema dell'ottimizzazione dei processi e di controllo, dove un modello di previsione è stato sviluppato per ciascun fase di lavorazione. Tuttavia, l'enfasi di questi progetti non è stata focalizzata sulla produzione programmata in quanto tengono conto solamente degli ordini dei clienti ma non delle previsioni sulla qualità.

### 1.3 Struttura della tesi

La tesi descrive tutte le fasi della realizzazione del software:

**Capitolo 2** Introduce il lettore alla descrizione del problema, della natura dei processi che sono stati modellati, e infine una discussione sugli aspetti innovativi introdotti dalla soluzione proposta; feedback sulla

qualità (utilizzo di indici definiti come KQI), inserimento delle richieste sulla qualità e infine un approccio all'ordinamento della produzione basata su un modello generale.

**Capitolo 3** Il terzo capitolo si occupa di illustrare la teoria che sta alla base delle scelte architettoniche del software sviluppato, ovvero a come si è deciso di trattare il problema introdotto nel secondo capitolo.

**Capitolo 4** In questo e nel prossimo capitolo viene descritto il lavoro svolto, più precisamente nel capitolo 4 viene esposta l'attività svolta per lo sviluppo della libreria che sta alla base della logica del simulatore. Viene inoltre presentata la parte relativa all'ottimizzazione.

**Capitolo 5** Continua a parlare del lavoro svolto, in questo capitolo viene presentata la progettazione e l'implementazione dell'interfaccia grafica che permette l'utilizzo della libreria sviluppata nel capitolo precedente.

**Capitolo 6** Una serie di test sono stati ideati ed effettuati per cercare di dimostrare la corretta implementazione del simulatore.

**Capitolo 7** Capitolo che riguarda le conclusioni e tratta i possibili futuri sviluppi.

# Capitolo 2

## L’impianto

La prima parte di questo capitolo introduce in maniera dettagliata la zona di impianto siderurgico di cui è stata realizzata la modellazione. Le varie entità che costituiscono l’area a freddo sono state esaminate e le caratteristiche essenziali sono state ricavate e utilizzate. È in questa area che il simulatore deve poter predire l’ordinamento dei coil (rotoli di lamina) da lavorare. In realtà il lavoro effettuato da queste entità non è di importanza fondamentale per lo modellazione del software, ma è necessario introdurre il lettore a queste componenti per migliorare la comprensione della natura del problema da risolvere. La seconda parte invece avvia il lettore alle problematiche introdotte dalla modellazione di questa area di impianto, per poi andare a spiegare come si è deciso di affrontare queste difficoltà e i caratteri distintivi del tipo di soluzione adottata.

### 2.1 Area di laminazione a freddo

L’area di laminazione a freddo rappresenta la zona all’interno dell’impianto in cui si incrudisce il materiale da lavorare, aumentandone la sua resistenza, ma con maggiore lavoro e dispendio energetico. Si riescono però ad ottenere così pezzi più rifiniti e precisi. La laminazione a freddo si esegue con la temperatura del metallo al di sotto della sua temperatura di ricristallizzazione (solitamente a temperatura ambiente), che aumenta la resistenza

tramite incrudimento fino al 20%. Inoltre, migliora la finitura superficiale e mantiene tolleranze più strette.

Comunemente i prodotti laminati a freddo comprendono lamiere, nastri e barre; questi prodotti sono solitamente più piccoli rispetto agli stessi prodotti che vengono laminati a caldo. A causa della ridotta dimensione dei pezzi e la loro maggiore forza, rispetto alla laminazione a caldo, sono utilizzati linee di produzione specificatamente progettate.

La figura 2.1 schematizza in maniera sintetica i componenti di quest'area, in particolare mostra i possibili percorsi che una striscia può intraprendere per ottenere il tipo di lavorazione desiderata.

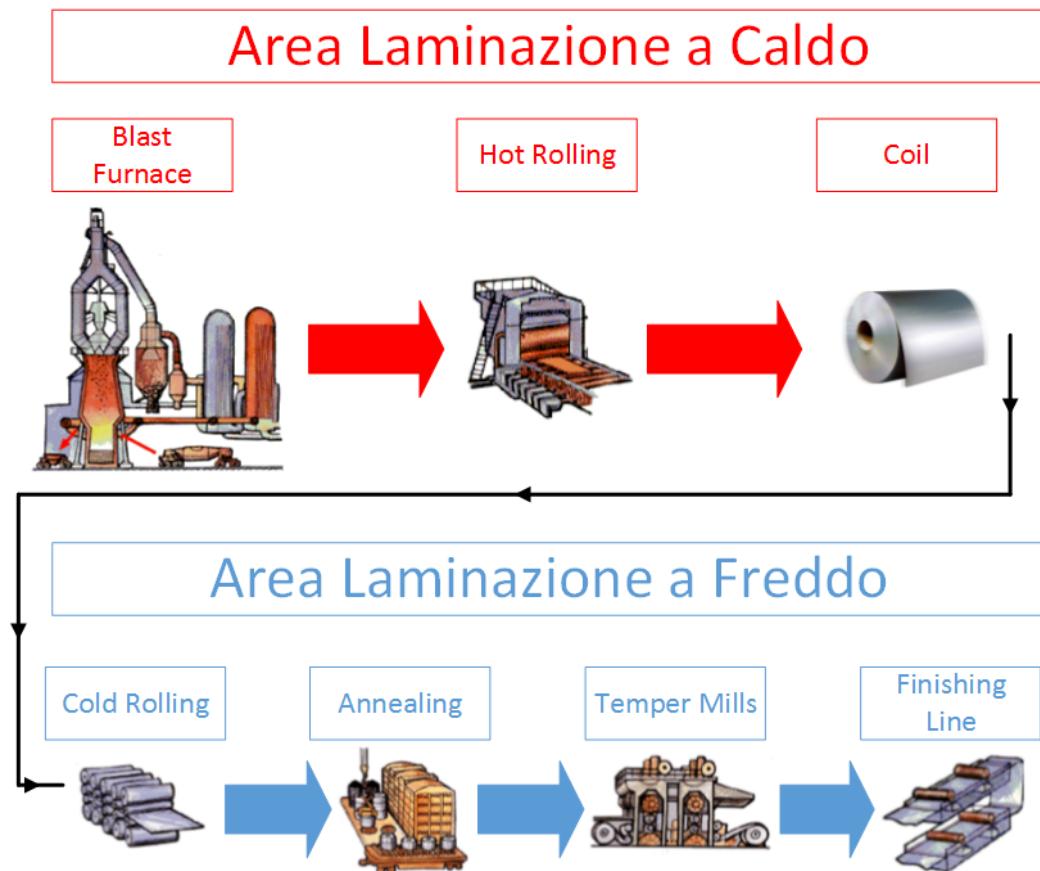


Figura 2.1: Diagramma globale dell'impianto

É giusto ricordare, visto che dalla figura non si evince, che ogni processo è composto da un numero variabile di linee, in cui ogni lavoro effettuato dallo step è replicato al suo interno. In realtà il numero di linee è stato deciso in fase di definizione delle specifiche di progetto, per poter impiegare il sistema su più impianti diversi è stato deciso di poter comporre l'impianto virtuale di un numero modificabile di linee appartenenti ad ogni step. Ogni processo è inoltre capace di stoccare una certa quantità di strip da lavorare; nei sotto paragrafi successivi segue una breve introduzione agli step della catena produttiva di questa area.

### 2.1.1 Cold Rolling

La laminazione a freddo rappresenta la prima fase di lavorazione per le strip in arrivo dalla hot rolling area (area di laminazione a caldo) il processo consiste nel far passare il materiale tra una coppia di

di rulli tenuti ad una distanza minore dello spessore del materiale in questo modo il semilavorato è costretto a ridursi di spessore (laminarsi) e si ottiene di conseguenza un allungamento nel senso di laminazione. Pur esistendo laminatoi che vengono utilizzati per materie plastiche artificiali e gomma, il campo metallurgico è comunque quello in cui tale tecnologia assume decisamente l'aspetto più rilevante. Per raggiungere lo spessore finale del materiale voluto può essere necessario eseguire più volte il processo

Figura 2.2: Schema del processo di laminazione

### 2.1.2 Annealing

La ricottura consiste nel riscaldamento ad una temperatura solitamente

inferiore a quella di fusione, seguito dalla permanenza di durata opportuna e da un lento raffreddamento rendendo così il materiale più dolce e più omogeneo. Nel caso dell'acciaio, si riscalda sino ad una temperatura poco superiore a quella di austenitizzazione<sup>1</sup> che ha un'importanza chiave nella tempra dell'acciaio e si mantiene a tale temperatura per un tempo sufficiente a trasformarlo completamente in austenite; segue poi un lento raffreddamento in forno. Tramite la ricottura viene alterata la microstruttura del materiale, causando mutamenti nelle sue proprietà quali la flessibilità e la durezza.

Il risultato tipico è la rimozione dei difetti della struttura cristallina. Può avere anche lo scopo di uniformare la composizione chimica dell'acciaio, in tal caso il riscaldamento è eseguito ad una temperatura più elevata e con tempi più lunghi.

### 2.1.3 Temper Mill

Un treno di rinvenimento è una processo di lavorazione composto da un passaggio orizzontale delle lamiere d'acciaio provenienti dalla laminazione a freddo, l'entrata e l'uscita di trasporto e di dispositivi a monte ed a valle variano a seconda del design e della natura del sistema di lavorazione. Lo scopo primario di un treno di rinvenimento è di migliorare la finitura superficiale sui prodotti siderurgici.



Figura 2.3: Processo di annealing, due blocchi da lavorare vengono riscaldati

---

<sup>1</sup>L'austenizzazione è una trasformazione di struttura cristallina degli acciai, consistente nell'ottenimento nell'acciaio stesso di una struttura austenitica, la quale è una soluzione solida di carbonio nel ferro



Figura 2.4: Output di una linea di temper mill

#### 2.1.4 Finishing Line

La linea di finitura è l'ultima postazione nel cammino di ogni rotolo, le finiture di acciaieria sia per i laminati a caldo che per quelli a freddo, sono la base delle condizioni di fornitura per tutti i prodotti piani di acciaio inossidabile. Sono comunemente utilizzate per la modifica delle superfici in modo da soddisfare le più diverse esigenze, le finiture eseguite hanno un'importanza diretta sull'aspetto della superficie, è possibile infatti scegliere tra una vasta gamma di finiture a seconda della superficie inox originale, del tipo e della trama dei nastri abrasivi<sup>2</sup> e delle spazzole, nonché della natura del procedimento di lucidatura usato. Esistono vari tipi di finitura superficiale nell'area di laminazione a freddo, tra cui quelle standard che si ottengono sul prodotto siderurgico direttamente in acciaieria durante la laminazione e i successivi cicli di trattamento.

Un piccolo elenco di lavorazioni di rifinitura dell'area a freddo sono:

**Finitura numero 2D** Effettua una riduzione di spessore di almeno il 20% sui nastri già laminati a caldo, la superficie non è ancora riflettente,

---

<sup>2</sup>La carta abrasiva, chiamata a volte anche carta vetrata serve per sgrossare, levigare e pulire superfici di legno o metallo

ma risulta levigata grazie alla laminazione a freddo subita. L'aspetto è grigio-argenteo, opaco e liscio.

**Finitura numero 2B** E' la finitura ottenuta mediante laminazione pellicolare a freddo, il suo aspetto è grigio-argenteo brillante ed è la finitura più diffusa per i laminati a freddo.

**Finitura BA** E' la finitura dei laminati a freddo ottenuta per trattamento termico di ricottura o ricristallizzazione.

## 2.2 Descrizione del problema

L'obiettivo finale è quello di ottimizzare il routing dei rotoli di acciaio sulle varie linee di produzione al fine di soddisfare indici di qualità definiti dagli ordini inoltrati dai clienti. In tal modo, le esigenze sulla qualità richiesta da parte del cliente influenzano la risultante filiera di produzione.

Con gli algoritmi di ottimizzazione, il massimo livello di qualità ottenibile può essere previsto per ogni rotolo e ogni percorso di produzione. In figura 2.5 la simulazione effettuata viene mostrata con un rotolo e due possibili clienti.

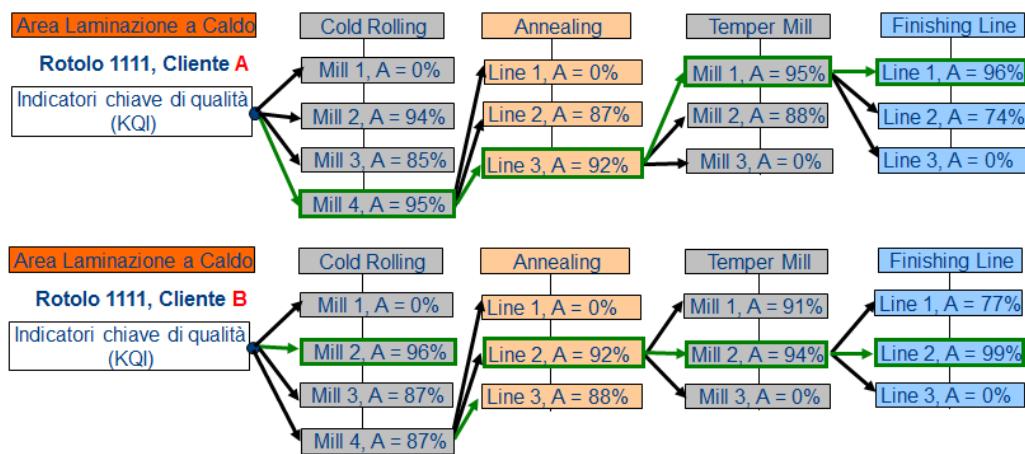


Figura 2.5: Percorso di produzione dipendente dalle previsioni sulla qualità e sulle esigenze di qualità del cliente

Il cliente A vuole produrre tubi e il cliente B intende produrre pannelli per il rivestimento di cucine. Entrambi i clienti che hanno effettuato richieste di qualità diverse per lo stesso tipo di acciaio:

- Il Cliente A ha bisogno di piccole tolleranze di spessore per la saldatura;
- Il cliente B richiede migliore qualità superficiale e di planarità per ragioni estetiche.

I modelli di simulazione danno una previsione sulla massima qualità ottenibile in termini di difetti di planarità e difetti di distorsione a balestra. Questi valori vengono utilizzati in una futura funzione di valutazione che considererà anche le esigenze individuali dei clienti. Il risultato è un valore di applicabilità A (0-100 %). Nell'esempio in figura 2.6, dove i percorsi per la migliore qualità per i clienti A e B sono segnati con rettangoli verdi.

Dal punto di vista qualitativo, il miglior percorso di produzione può essere determinato tramite la simulazione di tutte le combinazioni possibili. Ma questo non può essere l'unico criterio, perché poi le linee di produzione più performanti sarebbero sovraccaricate mentre le restanti linee resterebbero inutilizzate. Per superare questo inconveniente, verrà aggiunto un ulteriore indice di misura, il workload (carico di lavoro) di ogni linea.

La figura 2.6 visualizza il problema con più criteri di ottimizzazione in un esempio semplificato. Una certa quantità di rotoli deve essere prodotta (il rotolo con id 1111 e il rotolo con id 2222).

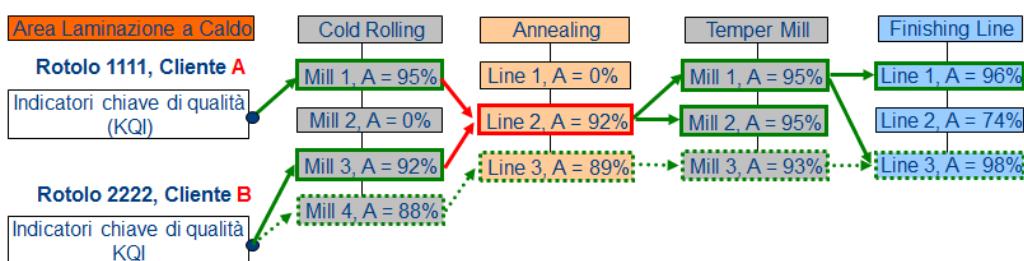


Figura 2.6: Ottimizzazione multi-criterio dei percorsi di produzione, prendendo in considerazione la qualità del processo e l'output di produzione

Alla fine della produzione nel hot strip mill (impianto di laminazione a caldo) gli indicatori chiave insieme alle raccomandazioni per l’ulteriore elaborazione per ogni rotolo, vengono determinati e utilizzati come parametri di input per l’ottimizzazione. Questa soluzione potrebbe portare ad un sovraccarico della linea 2 di ricottura e a perdite di tempo nella produzione, infatti, la linea 2 non potrebbe lavorare entrambi i rotoli nello stesso momento. Usando la capacità di lavorazione effettivamente utilizzata ovvero il workload come ulteriore criterio di ottimizzazione, la seguente alternativa segnata con linee verdi tratteggiate, potrebbe essere quella più vantaggiosa. Così, l’ottimizzazione trova il miglior compromesso tra il grado di utilizzo del processo e la qualità di produzione in uscita.

Nella situazione pratica, esistono molti altri criteri che devono essere inclusi nella funzione obiettivo, come la valutazione del difetto di distorsione a balestra e della planarità, che devono rispettare dei vincoli di tolleranza nella routine di ottimizzazione.

In conclusione i problemi di pianificazione della produzione possono essere molto difficili da risolvere in modo ottimale, soprattutto in impianti mediograndi, a causa della loro natura combinatoria (ci sono un gran numero di ordini, di itinerari possibili all’interno delle catene di produzione, di materiali, eccetera).

Al fine di includere i feedback sulla qualità nella pianificazione della produzione, dei KQI (key quality indicator, indicatori chiave di qualità) sono stati sviluppati in correlazione alla qualità ottenuta al termine di ogni fase di produzione e della qualità del prodotto complessivo. I KQI selezionati corrispondono ai tre indici di cui abbiamo appena parlato, ovvero i difetti di planarità, difetti di distorsione a balestra e il workload. Essi sono stati usati per la formulazione delle funzioni obiettivo basate sulla qualità e legate alla strategia di ottimizzazione.

Il BFI si è occupato della realizzazione indipendente dei modelli previsionali che forniscono i valori relativi ai KQI, le quali sono state integrate nell’applicazione di ottimizzazione della produzione. Le librerie sono in grado di predire i valori di qualità per i rotoli in uscita dalle linee che compongono i vari step dell’area di laminazione a freddo, in termini di difetti di

distorsione a balestra e planarità. I KQI sono stati impiegati in due modi diversi: i modelli di predizione forniti dal BFI e Outokumpu (gruppo industriale finlandese che opera nella produzione di acciaio inossidabile) sono stati impiegati nella simulazione dell'impianto per prevedere la qualità di un determinato rotolo al termine del suo percorso di lavorazione, mentre KQI effettivi saranno misurati on-line per mezzo di strumenti ad hoc e valutati dopo ogni fase di produzione per verificare l'effettiva qualità di un rotolo e per pianificare eventualmente il suo re-routing.

Per quanto riguarda l'ottimizzazione dei percorsi, è stato più conveniente talvolta non minimizzare una KQI, ma impostare un valore obiettivo o un intervallo di fattibilità per esso. Per questo motivo il problema di ottimizzazione è stato formulato utilizzando metodologie di programmazione a obiettivi, che consentono la trasformazione dei target in funzione obiettivo da minimizzare, come indicato nella tabella 2.1

Tipo	Target	Funzione Obiettivo
$\leq$	$f_i(\bar{x}) \leq t_i$	$\min\langle f_i(\bar{x}) - t_i \rangle$
$\geq$	$f_i(\bar{x}) \geq t_i$	$\min\langle t_i - f_i(\bar{x}) \rangle$
$=$	$f_i(\bar{x}) = t_i$	$\min f_i(\bar{x}) - t_i $
Range	$f_i(\bar{x}) \in [t_i^l, t_i^u]$	$\min(\max\{\langle t_i^l - f_i(\bar{x}) \rangle, \langle f_i(\bar{x}) - t_i^u \rangle\})$

Tabella 2.1: Traduzione dei target in funzioni obiettivo

dove l'operatore  $\langle z \rangle = \begin{cases} p(z), & \text{se } z > 0, \\ 0, & \text{altrimenti.} \end{cases}$  e  $p(z)$  è una funzione di penalizzazione. Passiamo ora ad analizzare le tre tipologie di KQI che sono stati selezionati.

### 2.2.1 Workload

Una funzione obiettivo molto importante riguarda il carico di lavoro relativo a ogni linea disponibile per ogni processo di lavorazione, come già accennato nel paragrafo 2.2.

Per evitare che vengano usate sempre le stesse linee per la lavorazione dei rotoli, si è cercato di distribuire in maniera più equa possibile il carico di lavoro tra le varie linee disponibili in ogni step. Per esempio, se nello step della laminazione a freddo sono presenti quattro linee, il simulatore cercherà di fare in modo che nel planning finale il carico di lavoro relativo ad ogni linea sia il più possibile vicino al 25%.

### 2.2.2 Crossbow

Il crossbow talvolta tradotta in balestra, è un difetto di produzione dei rotoli e indica la curvatura del rotolo di acciaio nel senso trasversale; il centro del rotolo con il difetto di distorsione a balestra è superiore ai bordi. Quando un rotolo viene lavorato per esempio durante la fase di laminazione, la parte sottoposta a pressione subisce come un'operazione di stretching; quando si verifica questo tipo di difetto le due superfici di area maggiore non sono più uguali tra di loro, ma una risulta maggiore dell'altra. L'area che aumenta di dimensione come nel caso dell'esempio mostrato in figura 2.7, si inarca verso il basso, mentre quella inferiore diminuisce e di conseguenza fa sì che il rotolo tenda ad arrotolarsi su se stesso lungo la linea che taglia longitudinalmente il pezzo.

Nei casi come il nostro in cui i pezzi posseggono un verso, la crossbow può risultare sia positiva (inarcamiento verso l'esterno) che negativa (inarcamiento verso l'interno).

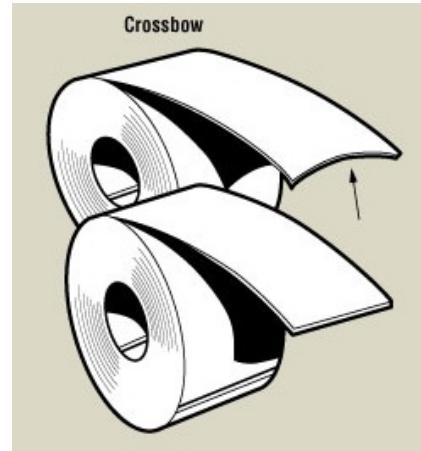


Figura 2.7: Esempio di rotolo affetto da crossbow

### 2.2.3 Flatness

Nel settore manifatturiero e meccanico, la planarità è una condizione geometrica importante per pezzi e utensili. Può sembrare un concetto molto

simile a quello di crossbow, ma di fatto non lo è. Nella fabbricazione di parti e componenti di precisione, in particolare nei casi in cui sono necessarie parti da collegare attraverso una superficie in maniera ermetica o a tenuta di liquido, la planarità è una qualità critica delle superfici prodotte.

Tali superfici sono lavorate o spianate per raggiungere il grado di planarità richiesta. In figura 2.8 viene mostrato un tipico caso di planarità non conforme.

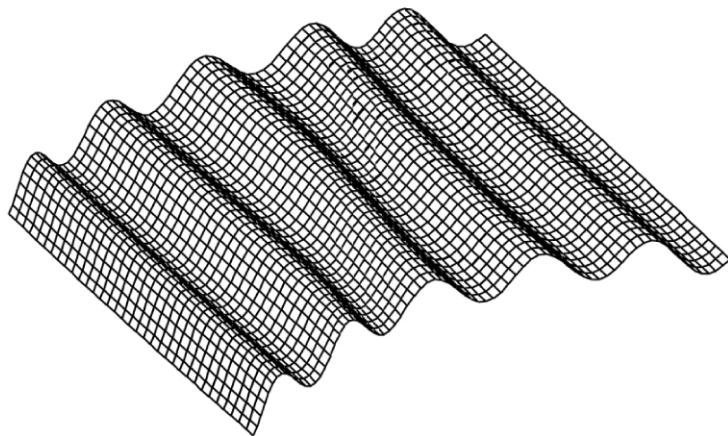


Figura 2.8: Rappresentazione di un pezzo non perfettamente spianato

La striscia in questione possiede dei sali e scendi lungo la sua superficie che non possono essere tollerati. Anche i valori assunti da questo parametro possono essere sia positivi che negativi.

### 2.3 Inclusione delle richieste sulla qualità nei requisiti per la pianificazione

Per quanto riguarda i KQI, anche le richieste dei clienti devono essere prese in considerazione. L'obiettivo principale è infatti quello di soddisfare tutti gli ordini dei clienti. E' stata eseguita una prima ottimizzazione della pianificazione all'inizio della zona di laminazione a freddo tenendo conto dei

reali KQI misurati dopo l'area di laminazione a caldo e quelli previsti. Alla fine del percorso simulato, i KQI previsti vengono abbinati con i KQI target definiti sulla base degli ordini dei clienti, come illustrato nella figura 2.1.

Più ordini clienti sono soddisfatti, meglio è.

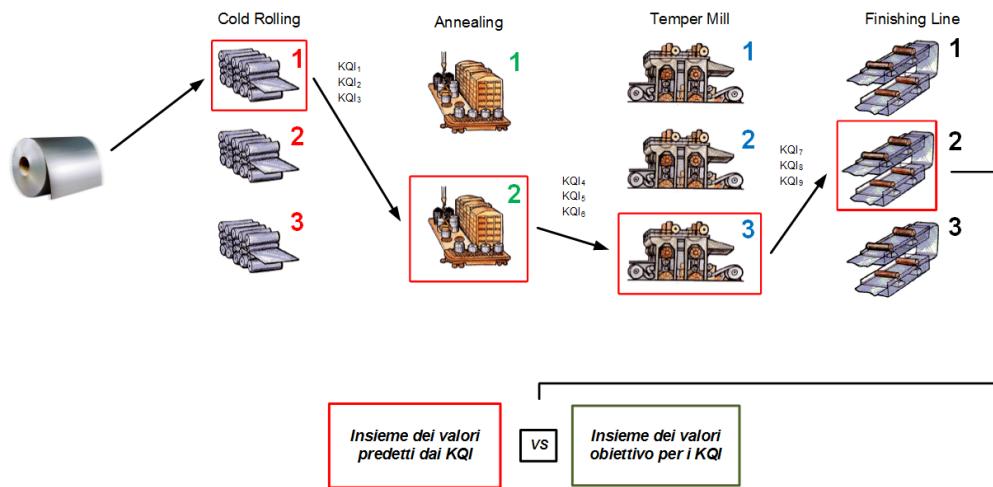


Figura 2.9: Processo di simulazione del percorso

La figura simula un percorso in cui all'uscita di ogni step vengono effettuate le valutazioni necessarie a quantificare il trattamento riservato alle strip in ingresso. Le valutazioni effettuate tramite KQI all'uscita di ogni step determinano la linea in cui effettuare la lavorazione successiva. Una volta terminato il percorso di simulazione, viene effettuato un confronto tra i valori previsti e i valori obiettivo per determinare se il percorso indicato è sufficiente a garantire una qualità maggiore o uguale a quella richiesta.

Come descritto nel paragrafo 2.2, è stato conveniente trasformare gli obiettivi relativi ai valori dei KQI in funzioni obiettivo da minimizzare.

## 2.4 Planning della produzione basato su un modello generale

Nella figura 2.10 è raffigurato il concetto di planning della produzione basato su un modello generale.

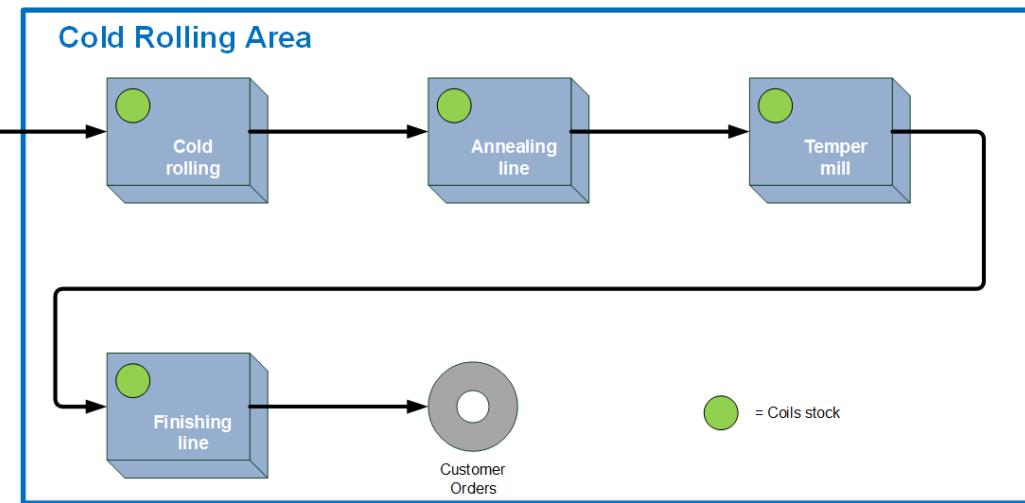


Figura 2.10: Diagramma dell'area di laminazione a freddo

Ogni rettangolo rappresenta una fase del processo, mentre i blocchi romboidali gialli rappresentano un blocco decisionale. Ad ogni blocco decisionale vengono eseguite le seguenti azioni:

1. KQI effettivi sono valutati mediante dati provenienti dal processo di lavorazione.
2. Simulazione di ulteriori fasi di processo.
3. Valutazione dei KQI previsti per mezzo di dati di simulazione.
4. Adeguamento del percorso di produzione pianificato.

Il primo blocco di decisione si trova alla fine della zona di laminazione a caldo. Sulla base dei KQI effettivi e predetti e sulla base di richieste dei

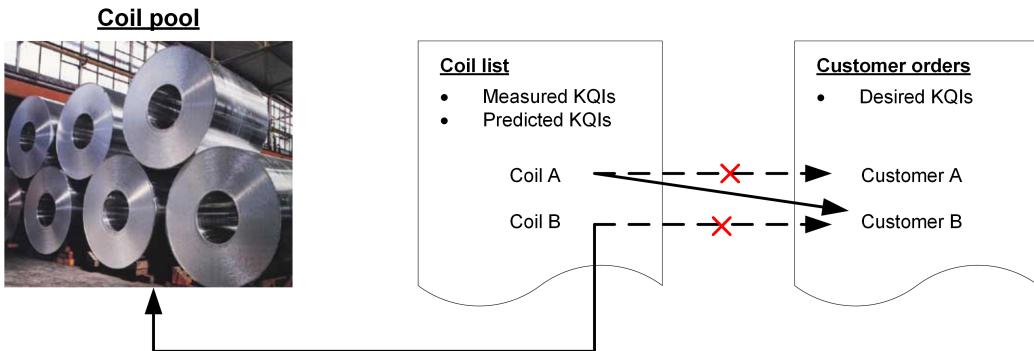


Figura 2.11: Possibile accoppiamento tra rotoli e ordini al termine della catena di processi

clienti, il percorso di produzione è ottimizzato. Dopo ogni fase di produzione nell'area di laminazione a freddo, i KQI reali vengono controllati per valutare se corrispondono con quelli previsti. Se si discostano più di una certa soglia, è necessario effettuare un adattamento del percorso di produzione che deve essere nuovamente pianificato.

A questo punto ci sono due possibili scenari: se ci sono sufficienti rotoli di alta qualità per soddisfare gli ordini con la domanda di qualità superiore, il rotolo può essere riassegnato. Altrimenti è necessario andare indietro nella catena dei processi per la fase di laminazione a freddo o alla linea di ricottura. Per questo motivo le fasi di processo nell'area laminazione a freddo sono caratterizzate anche dalla presenza di rotoli in stock che permettono alle bobine difettose provenienti dalle successive fasi di lavorazione di mettersi nella coda di attesa e di essere rielaborate. Alla fine della catena di processi ogni rotolo i cui difetti sono compatibili con le qualità ordinate, è assegnato a un ordine del cliente sulla base dei loro KQI effettivi finali. Se alcuni rotoli non soddisfano un qualsiasi ordine dei clienti sono inseriti in una coda di coil.

Consideriamo l'esempio raffigurato in figura 2.11, dove due rotoli, caratterizzati dai rispettivi KQI effettivi e KQI previsti, sono stati lavorati. Ci sono due ordini di clienti caratterizzati da alcune richieste sulla qualità del prodotto finale (KQI desiderati) e il primo (Cliente A) ha una richiesta di alta qualità. Inoltre, supponiamo che il rotolo A sia stato assegnato all'ordi-

ne del Cliente A nelle prime fasi del processo sulla base dei suoi KQI predetti. Eppure alla fine della catena di processi, i KQI misurati si discostano troppo dalle previsioni e non soddisfano più i requisiti di un cliente, diversi scenari possono verificarsi in questo punto, a seconda dei reali valori dei KQI relativi al rotolo B.

Per esempio, una soluzione possibile è quella di assegnare il rotolo A al Cliente B, se la richiesta è soddisfatta, e il rotolo B viene conservato in un lotto di rotoli dove possono essere presi rotoli per soddisfare nuovi ordini. Tuttavia, in questo caso un nuovo rotolo di alta qualità deve essere prodotto e quindi pianificato.

# Capitolo 3

## Problemi multi-obiettivo

In molte situazioni pratiche sorgono problemi di ottimizzazione con obiettivi multipli, dove gli obiettivi perseguiti sono in conflitto tra loro.

Un esempio concettualmente semplice per illustrare il tipo di problema è il seguente: una compagnia aerea vuole scegliere quale tipo di aereo da trasporto acquistare tra cinque possibili candidati e a tal fine vengono individuati quattro ragionevoli attributi rilevanti (chiamati anche funzioni obiettivo):

- Velocità massima (Km/h)
- Raggio di azione (Km)
- Carico massimo trasportabile (Kg)
- Costo (M€)

L'obiettivo globale della compagnia sarà quello di riuscire a spendere il meno possibile (minimizzazione dei costi) ed ottenere velivoli che abbiano valori per gli attributi che rientrino in determinati intervalli scelti in base alle esigenze della compagnia. Questi obiettivi possono essere in conflitto tra loro come per esempio è ragionevole pensare che un aereo che vada più veloce abbia anche un costo maggiore. Bisognerà quindi trovare il giusto compromesso tra velocità di trasporto e costo di acquisto del velivolo; questo equilibrio andrà raggiunto cercando di bilanciare i valori target di ogni funzione obiettivo.

Un problema di ottimizzazione multi-obiettivo (MOP, Multi Objective Optimization Problem) può essere definito come segue:

$$\min \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x})), \quad (3.1)$$

$$\text{soggetto a } \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), \dots, e_m(\mathbf{x})) \leq \mathbf{0} \quad (3.2)$$

dove  $\mathbf{x} = (x_1, \dots, x_n) \in \mathbf{X}$ ,

dove  $\mathbf{x}$  è la soluzione del problema,  $\mathbf{f}(\mathbf{x})$  è il vettore obiettivo, cioè il vettore contenente i risultati delle diverse funzioni obiettivo, ed  $\mathbf{e}(\mathbf{x})$  è il vettore dei vincoli con  $\mathbf{X}$  lo spazio delle soluzioni.

Per MOP non banali, non è possibile individuare una sola soluzione ottimale, ma il processo di ottimizzazione deve cercare un intero set di cosiddette soluzioni ottime di Pareto, che formano il Fronte di Pareto nello spazio degli obiettivi. Tutte le soluzioni appartenenti all'insieme di Pareto hanno la proprietà di essere non-dominante. Una soluzione  $a$  domina una soluzione  $b$  (vedere figura 3.1) se e solo se per ogni funzione obiettivo  $f_i(a) \leq f_i(b)$  ed almeno per una funzione obiettivo questo confronto è stretto:

$$a \succ b \iff \forall i : f_i(a) \leq f_i(b) \cap \exists j : f_j(a) < f_j(b) \quad (3.3)$$

L'insieme di Pareto è formato da tutte le soluzioni trovate, per cui la relazione 3.3 non vale. Tuttavia, in molte applicazioni reali la definizione analitica delle funzioni obiettivo con vincoli potrebbe essere molto difficile e talvolta impossibile senza una estrema semplificazione del problema; così i metodi tradizionali di ricerca operativa potrebbero essere inapplicabili.

### 3.1 Algoritmi evolutivi multi-obiettivo

Al fine di affrontare tali MOP, alcuni modelli specializzati di calcolo evolutivo sono stati sviluppati e applicati con successo [1].

Gli algoritmi evolutivi multi-obiettivo (MOEA, Multi Objective Evolutionary Algorithms), sono algoritmi iterativi di ricerca che esplorano lo spazio delle soluzioni per mezzo di alcuni meccanismi che si ispirano alla biologia

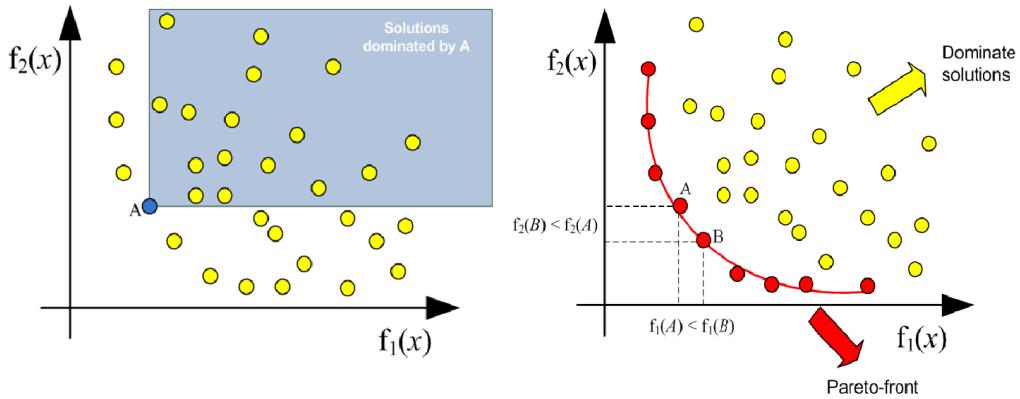


Figura 3.1: Esempi di dominanza (a sinistra) e del Fronte di Pareto (a destra) in una MOP di minimizzazione con due funzioni obiettivo

e all’evoluzione, come la mutazione e il crossover. Questa classe di modelli si basa su una popolazione, cioè l’insieme di soluzioni candidate, che evolve di generazione in generazione tramite l’accoppiamento o la mutazione. Un meccanismo di elitarismo<sup>1</sup> promuove quegli individui (cioè le soluzioni) che hanno miglior fitness (miglior valore per le funzioni obiettivo).

Le principali caratteristiche che rendono i MOEA più indicati per la soluzione di problemi multi-obiettivo reali sono:

- I MOEA possono approssimare un intero Fronte di Pareto in una singola esecuzione dell’algoritmo;
- I MOEA sono in grado di affrontare problemi con vincoli complessi;
- I MOEA possono essere inoltre utilizzati se il problema non può essere rappresentato in maniera analitica.

Quest’ultima ultima caratteristica si rivela la più interessante, infatti i MOEA possono essere utilizzati per simulatori o emulatori software, e sono in grado di sostituire la definizione matematica del problema.

---

<sup>1</sup>L’elitarismo consiste in un ulteriore passo precedente alla selezione che copia nelle nuove popolazioni anche gli individui migliori della popolazione precedente

Per esempio, la logistica di un impianto complesso con differenti linee di produzione, ciascuna delle quali con i propri vincoli, può essere più facilmente e realisticamente rappresentato da un simulatore piuttosto che da delle equazioni. Per queste ragioni nell'ambito del progetto TECPLAN sono state impiegate queste tecniche. Qualsiasi algoritmo evolutivo è basato sui seguenti passi:

1. Generazione di una popolazione iniziale casuale costituita da cromosomi (soluzioni candidate);
2. Valutazione della funzione di fitness (punteggio) di ogni cromosoma, basata sulle funzioni obiettivo e la Dominanza di Pareto come espressa in 3.3;
3. Selezione dei cromosomi scelti per l'accoppiamento;
4. I cromosomi scelti vengono divisi in due gruppo che saranno soggetti all'operazione di crossover o mutazione;
5. Tornare al secondo step finché qualche criterio di stop non viene raggiunto.

Così, ad ogni generazione (iterazione), l'algoritmo stocasticamente conduce la popolazione verso le migliori soluzioni. Inoltre, ad ogni iterazione, il miglior cromosoma può essere salvato, introducendo il cosiddetto meccanismo di elitarismo. Al termine dell'ultima generazione la popolazione d'élite rappresenta l'approssimazione del Fronte di Pareto.

Tuttavia, esistono diversi paradigmi di MOEA che differiscono soprattutto nel modo di affrontare l'esplorazione del Fronte di Pareto, per esempio in [3]. Gli algoritmi evolutivi multi-obiettivo più comunemente utilizzati sono:

- Niced Pareto Genetic Algorithm (NPGA) [5]
- Non Dominated Sorting Genetic Algorithm II (NSGA-II) [7]
- Strength Pareto Evolutionary Algorithm 2 (SPEA2) [4]

Il laboratorio PERC.RO. ha messo a disposizione per lo sviluppo del simulatore una propria implementazione del secondo e terzo paradigma, in attesa di ottenere un'implementazione anche del primo.

## 3.2 Algoritmi Genetici

Una breve introduzione agli algoritmi genetici (noti anche come GA, Genetic Algorithm) è necessaria per la comprensione del ruolo svolto dalle funzioni implementate durante lo sviluppo dell'applicativo.

Gli algoritmi genetici rientrano nella classe delle euristiche, ovvero quegli algoritmi che non garantiscono di ottenere la soluzione ottima, ma tendono ad essere particolarmente efficaci nell'esplorazione di varie porzioni della regione ammissibile e nell'evolvere gradualmente verso soluzioni sempre migliori. Questa strategia è motivata dal fatto che gli algoritmi euristici vengono solitamente impiegati quando un metodo di risoluzione esatto non è possibile o sarebbe proibitivo in termini di tempo computazionale, accontentandosi così di ricercare una soluzione sub ottimale in un tempo ragionevole. Essi sono procedure complesse, adattative, finalizzate alla risoluzione di problemi di ottimizzazione e basate concettualmente sui principi che regolano l'evoluzione della specie. Tale tipologia di algoritmi è stata introdotta da John Holland [6] e da allora sono stati sviluppati e applicati ad un insieme estremamente vasto di problemi in svariati ambiti [2]. I risultati di queste ricerche hanno mostrato l'elevato grado di robustezza di questa metodologia rispetto a quelle già presenti in letteratura.

Gli algoritmi genetici<sup>2</sup> riproducono il processo evolutivo delle specie animali, considerando una popolazione di cromosomi (detti individui) che rappresentano le possibili soluzioni del problema. La funzione di fitness, in un certo senso, indica l'adattabilità all'ambiente: gli individui che meglio si adattano hanno più probabilità di riprodursi e di trasmettere i propri geni alle generazioni future. La qualità di un individuo (cioè quanto è buona la

---

<sup>2</sup>Introduzione agli algoritmi genetici della Professoressa Beatrice Lazzerini, Professore Ordinario presso il Dipartimento di Ingegneria dell'Informazione dell'Università di Pisa, <http://info.iet.unipi.it/~lazzerini/icse/LucidiGenetici.pdf>

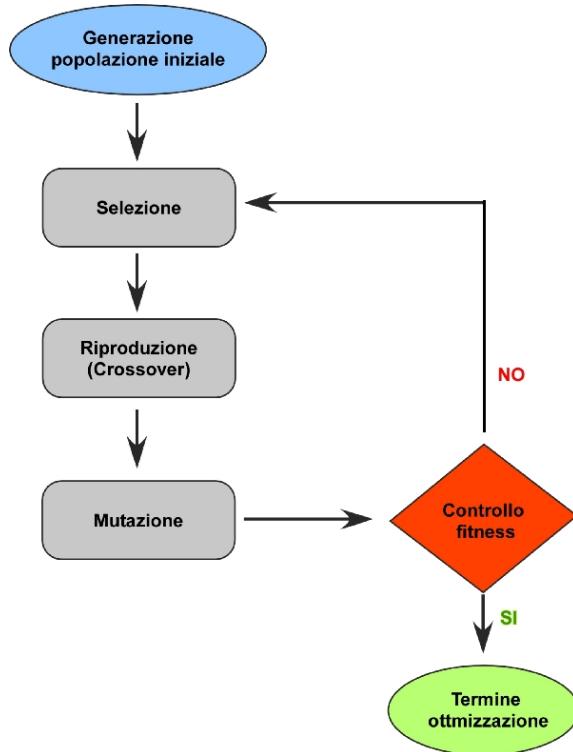


Figura 3.2: Diagramma di flusso di un generico algoritmo genetico

soluzione del problema) è misurata mediante la funzione di fitness. Un algoritmo genetico è una procedura di ricerca iterativa il cui scopo è quello di ottimizzare la funzione di fitness; partendo da una popolazione iniziale, un GA produce nuove generazioni che contengono individui migliori delle precedenti.

L'algoritmo evolve così verso l'ottimo globale della funzione di fitness, in figura 3.2 viene illustrata la sequenza di azioni compiute dall'esecuzione di un generico GA. I nomi di funzioni che si trovano all'interno di un rettangolo vengono sempre eseguite una volta ricevuto l'input, l'unica funzione che svolge un compito decisionale in base all'input ricevuto, è la routine di fitness (valutazione).

In realtà non è garantito che un GA trovi una soluzione ottima globale; un algoritmo genetico è in grado di trovare soluzioni buone in tempi ragionevoli.

Nel modello tradizionale, i cromosomi sono stringhe di bit di lunghezza

fissa e tutte le generazioni hanno la stessa dimensione (numero di individui), ogni cromosoma rappresenta un punto nello spazio di ricerca. I più importanti operatori di ricerca sono la riproduzione (o crossover) e la mutazione. Con l'operazione di crossover, i geni di due individui, vengono ricombinati per produrre individui figli che ereditano caratteristiche da entrambi i genitori; mentre con la mutazione si reintroduce nella popolazione materiale genetico andato perduto con l'operazione di crossover.

La ricerca genetica realizza un compromesso tra exploitation (sfruttamento) della soluzione disponibile migliore ed exploration (esplorazione) dello spazio di ricerca. Exploitation ed exploration corrispondono, rispettivamente, a ricerca locale e ricerca globale: exploitation eccessiva può portare l'algoritmo a convergere ad una soluzione sub-ottimale (la ricerca resta intrappolata in un ottimo locale), exploration eccessiva può non sfruttare appropriatamente la conoscenza già disponibile rendendo il processo di ricerca molto lento.

### 3.3 Codifica del cromosoma

Un cromosoma rappresenta una possibile soluzione al problema multiobiettivo, cioè una rappresentazione genetica del dominio delle soluzioni. Nel nostro caso esso rappresenta un possibile planning di tutti i rotoli da lavorare, la sua lunghezza infatti è proporzionale proprio dalla quantità di rotoli da lavorare.

Soltanente il cromosoma viene codificato tramite l'uso di array di bit, ma nella corrente implementazione, il tipo di contenuto del vettore è ricaduto su valori interi, sufficienti a identificare le linee appartenenti ai vari step che compongono l'area di laminazione a freddo. Ogni elemento del cromosoma viene chiamato gene, mentre l'insieme di quattro geni consecutivi è stato definito come macrogene. Ogni macrogene rappresenta una possibile rottura all'interno dell'area di laminazione a freddo, ed è infatti composto da quattro geni, uno per ogni fase di lavorazione all'interno della catena; il numero di macrogeni è proporzionale al numero dei rotoli di cui è richiesta la lavorazione e non può essere modificato.

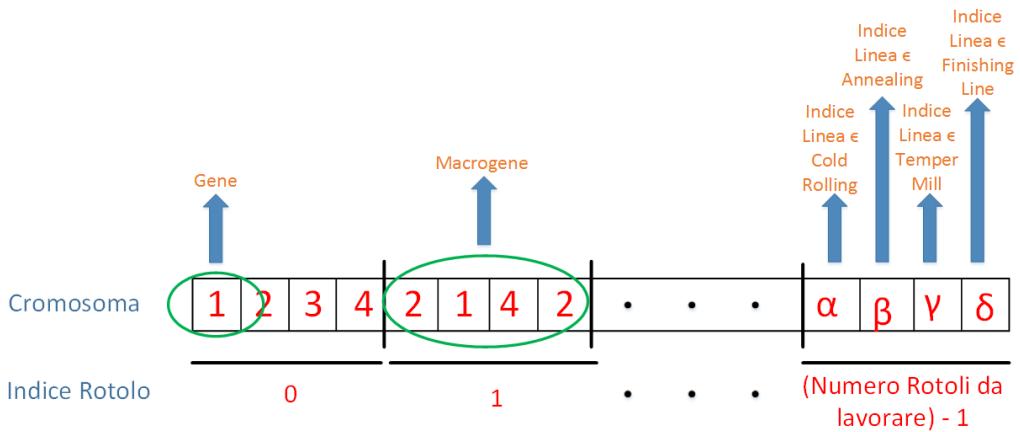


Figura 3.3: Codifica del cromosoma scelta, identificazione del macrogene, del gene, e del significato assunto da ognuno di essi

L’insieme di valori che ogni gene può assumere dipende dal numero di linee appartenenti allo step di lavorazione, il quale viene identificato dall’indice interno al macrogene; se 0 corrisponde al processo di laminazione a freddo il 3 corrisponde alla linea di rifinitura. Per esempio se si guarda al valore di indice 2 all’interno del macrogene, tale valore si riferisce alla linea utilizzata per il processo di temper mill per quel rotolo. Ogni volta che si genera un cromosoma, esso determina univocamente per ogni rotolo da lavorare il suo percorso all’interno della catena produttiva.

Dalla figura 3.3 si nota come per il rotolo con numero identificativo pari a 1 si sia deciso di passare dalla linea 2 per il processo di cold rolling, per la linea 1 nel processo di annealing, per la linea 4 per lo step di temper mill ed infine per la linea 2 del processo di finishing line.

### 3.4 Funzione di Creazione

La funzione di creazione genera casualmente una prima popolazione di cromosomi (soluzioni), permettendo la generazione di una vasta combinazione di possibili soluzioni (spazio di ricerca) tramite l’utilizzo di una matrice di interi. In base alle prove effettuate si è rivelato sufficiente creare una po-

polazione iniziale di 50 individui, al termine dell'esecuzione del GA, saranno disponibili 25 soluzioni da scegliere per l'operatore.

Per ogni gene appartenente ad ognuno dei cromosomi è bastato generare un numero casuale in base al numero di linee che compongono il processo.

## 3.5 Funzione di Crossover

La routine di crossover genera una nuova soluzione mescolando tra loro il contenuto di due soluzioni esistenti, le metodologie più comunemente utilizzate sono:

**Crossover a un punto** È la tecnica più semplice, date due soluzioni viene individuato un punto che separa ciascuna di esse in due parti, una chiamata testa e una chiamata coda. In seguito la nuova soluzione verrà composta dalla testa di una soluzione e dalla coda di quella rimanente.

**Crossover a due punti** Consiste nel considerare due soluzioni adatte all'evoluzione e nel tagliare i loro vettori di codifica in due punti predefiniti o casuali al fine di ottenere una testa, una parte centrale ed una coda estratti dalla prima e dalla seconda soluzione. La prima nuova soluzione sarà data dalla testa e della coda della prima soluzione e dalla parte centrale della seconda soluzione. La seconda nuova soluzione sarà data dalla parte centrale della prima soluzione e dalla testa e dalla coda della seconda soluzione.

**Crossover uniforme** Consiste nello scambiare casualmente il contenuto dei geni tra le soluzioni candidate all'evoluzione.

Tra le varie opzioni disponibili, si è deciso di utilizzare l'operatore di crossover uniforme, quella più semplice e concettualmente più utile ai nostri scopi in quanto i valori che vengono ereditati dal figlio sono sempre proporzionali al numero di linee utilizzate in un determinato step. Il crossover a due punti non garantisce tale proprietà.

Prese due soluzioni candidate, chiamate una soluzione padre e l'altra soluzione madre, per ogni gene viene casualmente generato un valore di pro-

babilità compreso tra 0 e 1. Se il valore generato è minore di 0,5 allora la soluzione figlio prenderà il valore del gene padre, altrimenti il valore del gene madre. Questa operazione viene eseguita per ogni gene del figlio (che possiede lo stesso numero dei geni del padre e della madre), e permette a partire da un mating pool (insieme d'accoppiamento) di ottenere una nuova popolazione in cui tutte le nuove soluzioni generate hanno qualcosa in comune con le precedenti generazioni. In un mating pool un individuo può comparire anche più volte, le coppie generate possono essere scelte in base a diverse strategie di selezione (roulette wheel, tournament selection, steady-state selection, eccetera), queste strategie dipendono dall'implementazione specifica del GA.

## 3.6 Funzione di Mutazione

La funzione di mutazione è un operatore genetico, utilizzato da una generazione di una popolazione di cromosomi a quella successiva, per aumentare la diversità genetica ed è quindi analoga alla mutazione biologica.

La mutazione altera uno o più valori dei geni in un cromosoma, a partire dal suo stato iniziale. Con la mutazione, la soluzione può cambiare completamente dalla soluzione precedente, quindi un GA può generare una soluzione migliore utilizzando la routine di mutazione. Cosa importante, la mutazione avviene durante l'evoluzione secondo una probabilità di mutazione prefinita. Questa probabilità viene impostata a una valore basso, altrimenti, se troppo alto, la ricerca si trasformerà in una primitiva ricerca casuale.

Per il progetto è stato deciso di realizzare una funzione dedicata; per ogni soluzione viene generato un valore di probabilità casuale compreso tra 0 ed il numero dei rotoli da lavorare. Successivamente, in base a quel valore viene individuato uno dei macrogeni appartenenti al cromosoma; per il macrogene selezionato, vengono generati quattro valori casuali, uno per ogni gene. Questi valori andranno a sostituire completamente il contenuto del macrogene; ognuno di questi valori (geni) ha naturalmente un range di possibili valori che rientra nell'intervallo disponibile per quel processo di lavorazione.

Quindi, solo uno dei macrogeni all'interno della soluzione verrà modificato in maniera completa lasciando i restanti inalterati.

## 3.7 Funzione di Fitness

Una funzione di fitness è un particolare tipo di funzione obiettivo utilizzata per riassumere, come un indice di merito, quanto una determinata soluzione riesca a conseguire gli scopi definiti. Nel caso del simulatore, la funzione di fitness deve valutare l'insieme di soluzioni proposte per controllare se una o più soluzioni rispettano i vincoli imposti sui KQI.

Se tutte le soluzioni soddisfano i vincoli oppure il numero massimo di iterazioni è stato raggiunto, l'algoritmo termina. Altrimenti, l'algoritmo riprende la sua esecuzione applicando le varie funzioni basate sulla selezione naturale alle soluzioni che non hanno soddisfatto i criteri.

# Capitolo 4

## Sviluppo Libreria

Questo capitolo è dedicato alla descrizione del lavoro svolto, le scelte strutturali e i paradigmi di programmazione utilizzati per la realizzazione del software di simulazione. Questo capitolo è la guida che uno sviluppatore seguirà nel caso volesse portare avanti tale lavoro, per capire quali sono gli elementi coinvolti nel raggiungimento di ogni singolo task progettuale. Il simulatore è stato sviluppato sotto forma di libreria DLL (Dynamic-Link Library), una libreria dinamica altro non è che un file che contiene del codice, al quale si accede effettuando precedentemente un collegamento dal punto di vista del software che ne farà uso.

Nel caso di una libreria dinamica ciò avviene per via esplicita, il collegamento avviene esclusivamente a runtime, cioè in fase di esecuzione in maniera tale che possa essere utilizzata dall'interfaccia grafica descritta nel capitolo 5.

### 4.1 Ambiente di sviluppo

Le specifiche definite dal work package all'interno del progetto TECPLAN hanno imposto di realizzare la libreria utilizzando come linguaggio di programmazione C++. Il C++<sup>1</sup> è un linguaggio di programmazione orientato agli oggetti, con tipizzazione statica. È stato sviluppato (in origine col nome

---

<sup>1</sup><http://it.wikipedia.org/wiki/C%2B%2B>

di C con classi) da Bjarne Stroustrup ai Bell Labs nel 1983 come un miglioramento del linguaggio C. Tra i miglioramenti principali troviamo: l'introduzione del paradigma di programmazione a oggetti, funzioni virtuali, overloading degli operatori, ereditarietà multipla, template e gestione delle eccezioni.

Come ambiente di sviluppo integrato è stato messo a disposizione da parte del SIAD l'editor Microsoft Visual Studio 2008, che ha permesso inoltre lo sviluppo dell'interfaccia in linguaggio C#. Sempre la divisione SIAD ha anche messo a disposizione librerie contenenti due tra le implementazioni possibili per gli algoritmi genetici; NSGA-II e SPEA2.

Inoltre altre librerie che mettono a disposizione strutture dati come matrici e vettori sono state utilizzate insieme ad ulteriori librerie matematiche comprendenti funzioni statistiche; entrambe le tipologie di DLL sono sempre state fornite e sviluppate da parte del SIAD.

## 4.2 Pattern architetturale

La grande mole di lavoro da svolgere richiesta dallo sviluppo dell'applicazione, ha posto il problema di come decomporre il progetto in parti più piccole, in maniera tale da poter suddividere il lavoro in varie tappe; così facendo è stato possibile rendere incrementale lo sviluppo del codice. Un'altra necessità è stata quella di voler suddividere in diversi componenti la struttura del programma ed è per questo che è stato scelto il pattern<sup>2</sup> MVC (Model-View-Controller). Il pattern MVC, nato nei laboratori Xerox negli anni'70 all'alba dell'era delle interfacce grafiche, ha dato origine a molti pattern derivati e ad un enorme numero di implementazioni, venendo integrato in diversi tra i framework più usati come JEE, .NET, Ruby.

Il pattern architetturale MVC fu introdotto allo scopo di facilitare lo sviluppo di applicazioni composte da numerose interfacce utente, ciascuna caratterizzata da una sua logica di interazione e di trasformazione dei dati trattati. La necessità di gestire la complessità che si accompagnava ad uno

---

<sup>2</sup>Nella programmazione orientata agli oggetti del software, un design pattern è una soluzione progettuale a un problema ricorrente, la parola design viene spesso interpretato come modello

sviluppo del tutto destrutturato, caratterizzato per lo più da blocchi monolitici di codice poco specializzato, condusse alla ricerca di un approccio alternativo che garantisse una migliore organizzazione del codice sorgente.

Il pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

**Model** è l'insieme dei componenti che mantengono lo stato, ossia i dati e i metodi per accedervi. Costituisce la logica applicativa.

**View** deputato alla visualizzazione vera e propria dell'interfaccia utente per la presentazione dei dati.

**Controller** gestisce le interazioni dell'utente con l'applicazione (tipicamente intercettando e gestendo gli eventi e gli input dell'utente), mediante accesso al model e definendo la view corrispondente da presentare.

In figura 4.1 vengono illustrate le relazioni tra i vari componenti con commenti sui ruoli interpretati.

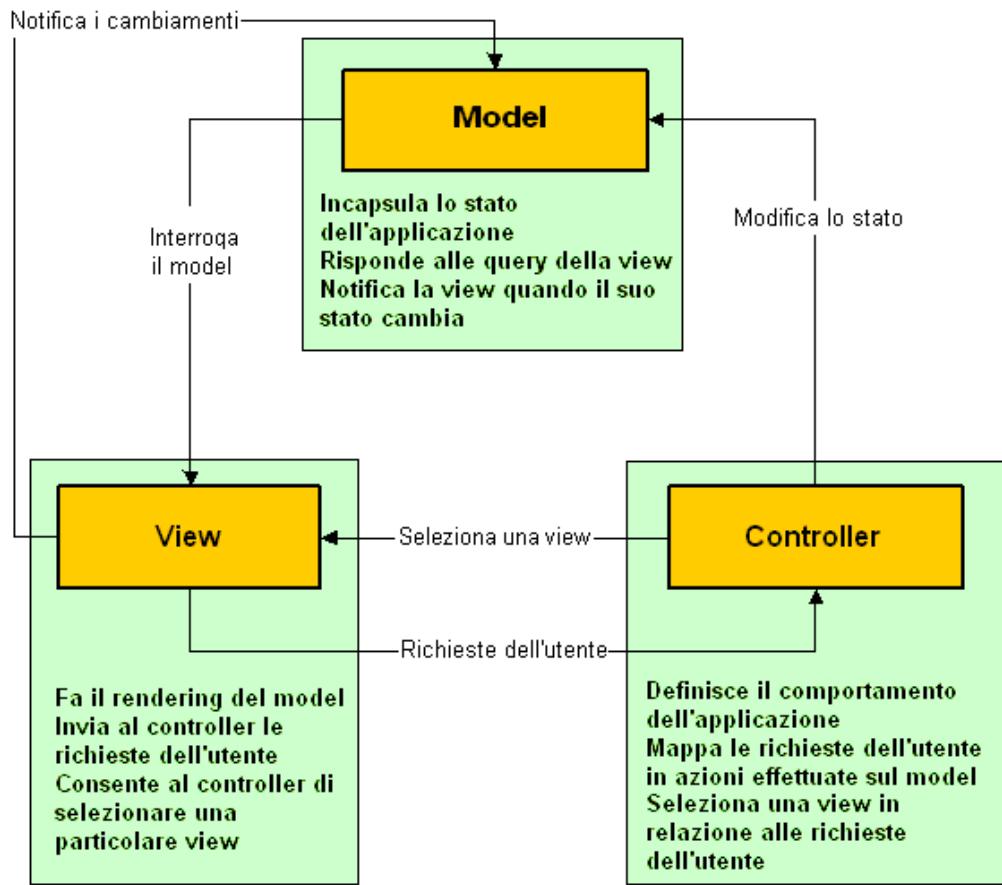


Figura 4.1: Pattern MVC puro

La struttura del software benché si ispiri fortemente a questo paradigma presenta notevoli differenze, in particolar modo nel ruolo del controller rispetto agli altri componenti: il suo è un canale di comunicazione esclusivo nei confronti dell'interfaccia:

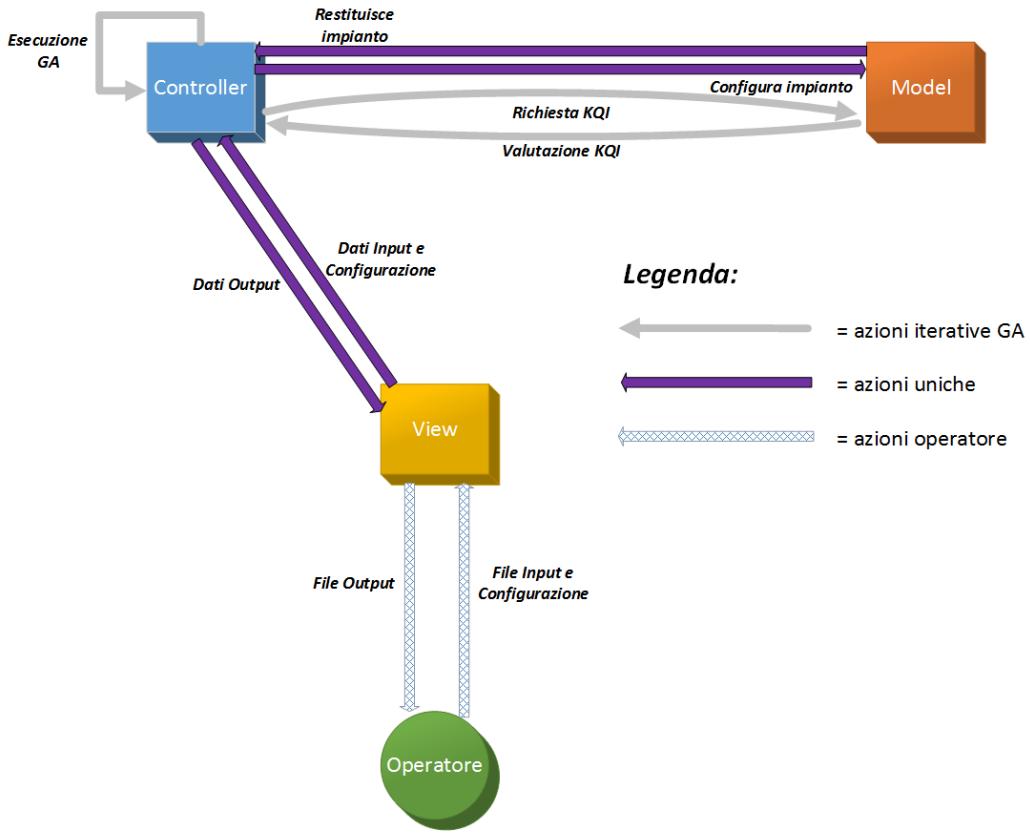


Figura 4.2: Variante pattern MVC

A differenza del pattern standard in questa implementazione la view non interroga il model ma si occupa esclusivamente di passare i dati di input al controller. Una volta terminata la simulazione il controller ottiene i risultati sulla pianificazione dal model che azzera il suo stato, il controller consegna quest'ultimi alla view per mostrarli all'utente. Inoltre la view è composta da un'unica schermata, il che semplifica la gestione della GUI da parte del controller. Un altro ruolo del controller è di gestione dell'esecuzione del GA, per fare ciò interroga il model che restituisce al GA i valori predetti per i KQI selezionati. Il controller oltre ad occuparsi della gestione dei modelli, costruisce l'istanza globale dell'impianto. Il model corrisponde a componenti come i processi produttivi, le linee che appartengono ai processi e le librerie dei modelli che generano previsioni per i valori dei KQI.

### 4.2.1 Modelli

I modelli usati si dividono in due categorie: quelli che astraggono i componenti principali della catena di lavorazioni, e quelli utilizzati per il calcolo dei KQI, iniziamo descrivendo il primo tipo:

**ProcessingStep** rappresenta un processo produttivo generico della catena di lavorazioni tra i quattro processi presenti nell'area di laminazione a freddo: laminazione a freddo, ricottura, treno di rinvenimento e linea di rifinitura.

**ProcessingLine** astrae una linea appartenente a un generico processo, le specifiche non hanno richiesto di aumentare il grado di dettaglio; in pratica un oggetto di questo tipo può appartenere a qualsiasi tipo di ProcessingStep.

Tramite questi due modelli è possibile effettuare l'astrazione dell'area di laminazione a freddo.

La seconda categoria riguarda i modelli utilizzati per il calcolo dei KQI che a loro volta si suddividono in due tipologie:

**LineKQI** rappresenta un KQI che può essere calcolato direttamente dall'oggetto che rappresenta la linea, come il workload.

**PlantKQI** delinea un KQI definibile come globale ovvero che viene computato sul prodotto finale in uscita dall'ultimo processo produttivo ovvero la linea di rifinitura.

Con questi due tipi di modelli ad ogni iterazione i GA possono valutare le soluzioni proposte tramite la funzione di fitness.

## 4.3 Schema generale

Lo schema delle classi viene utilizzato per descrivere il livello di astrazione del contesto dopo aver effettuato l'analisi del dominio che in questo caso corrisponde all'area di laminazione a freddo.

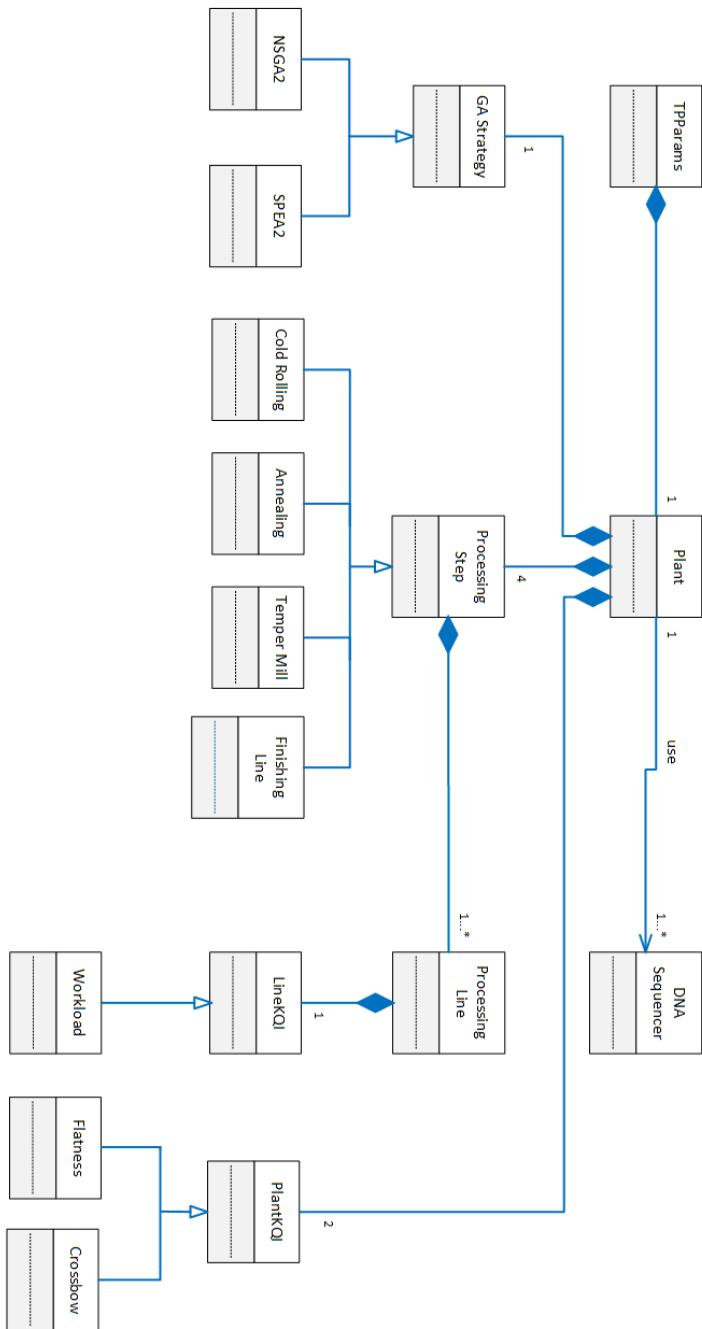


Figura 4.3: Schema delle classi della libreria

La classe `Plant` rappresenta l'impianto al cui interno si trova l'area di laminazione a freddo composta da quattro processi `ProcessingStep`, ognuno

specializzato come Cold Rolling, Annealing, Temper Mill o Finishing Line). Ogni processo è composto da varie linee ProcessingLine che non presentano differenze l'una dall'altra in base al processo di appartenenza. Ogni ProcessingLine possiede un LineKQI (KQI relativo alla linea). Anche in questo caso è stato usata il polimorfismo per specializzare il tipo di KQI di linea; potranno così essere aggiunti ulteriori tipi di KQI. L'impianto è composto anche da KQI globali ovvero che vengono utilizzati per prevedere la qualità dei semilavorati in uscita dalla linea di rifinitura che rappresenta l'ultimo processo di lavorazione. Inoltre Plant usa la classe DNASequencer per codificare i cromosomi che fanno parte dei GA. TPPParams mette a disposizione una serie di funzioni tramite cui la GUI può passare i parametri presi in input alla libreria.

Una volta eseguita la simulazione un'insieme di funzioni esportate viene messa a disposizione per poter ricavare i dati di output dalla libreria; tramite esse è possibile recuperare:

**Fronte di pareto** che è composto da un'insieme di vettori contenenti i valori medi dei KQI selezionati per ogni soluzione proposta dal GA.

**Insieme di pareto** composto da un'insieme di vettori, ogni vettore rappresenta soluzione del problema.

**Lista dei matching** per ogni soluzione appartenente dell'insieme di Pareto è associato un vettore che contiene la lista degli accoppiamenti tra rotoli e ordini.

**Linee attive** Numero di linee attive come viene specificato nel file di configurazione.

#### Tipi di KQI valutati per ogni step

##### 4.3.1 Controller

Il ruolo di controller viene esercitato da un'istanza della classe Plant la quale ha il compito di astrarre l'area di laminazione a freddo e dalla classe TPPParams che definisce il punto di ingresso per l'applicazione . I modelli di

KQI, sia di linea che di impianto sono contenuti all'interno dell'istanza. Il controller si occupa di leggere i file di input contenenti la lista dei coil da lavorare, la lista degli ordini e il file di configurazione dell'impianto. Altro ruolo importante del controller è quello di contenere le funzioni implementate per i GA descritte nel capitolo 3.2. La classe Plant contiene l'algoritmo di ottimizzazione che fornisce il matching migliore tra rotoli e ordini per l'insieme di soluzioni fornita dai GA.

La classe TPPParams mette a disposizione una serie di funzioni tramite cui la view può effettuare il passaggio dei parametri di input alla libreria, per poter poi prelevare i dati di output forniti dal GA. Per il passaggio dei dati in ingresso è stata definita la routine:

```
int optimize(int GenAlgType, const char* configFile,
           const char* coilsOrderFile, const char* coilsListFile)
```

La chiamata a questa funzione fa sì che venga eseguito il *main()* principale il quale esegue in sequenza:

1. Crea un'istanza dell'impianto;
2. Configura l'impianto attraverso la lettura del file di configurazione il cui path assoluto corrisponde a *constchar \*configFile*;
3. Legge il contenuto del file comprendente gli ordini da soddisfare in *constchar \*coilsOrderFile*;
4. Legge la lista dei coil disponibili in *constchar \*coilsListFile*;
5. Seleziona il tipo di GA desiderato tramite *intGenAlgType*;
6. Esegue il GA.

Una volta che il GA termina, vengono restituite tre strutture dati contenenti il fronte di Pareto, l'insieme di Pareto, e la lista degli accoppiamenti tra rotoli e ordini. La view può ottenere i dati in output dall'algoritmo tramite le restanti funzioni di export offerte dalla libreria:

**void getParetoFrontElem(int frontIdx, int\* nData, double\*\* pData)**

serve a prelevare una riga della matrice contenente il fronte di pareto, la riga di indice *idx* viene copiata nel all'interno dell'area di memoria puntata da *pData* un array di double, in *nData* conterrà la lunghezza del vettore copiata.

**void getParetoSetElem(int frontIdx, int\* nData, double\*\* pData)** il comportamento è analogo a quello di *getParetoFrontElem()* ma all'interno di *pData* viene copiata la riga di indice *idx* della matrice contenente l'insieme di pareto.

**void getMatchesElem(int solutionIdx, int\* nData, double\*\* pData)**

il modo di agire è lo stesso delle due funzioni precedenti con la differenza che *pData* conterrà la riga di indice *idx* della matrice che contiene la lista di matching tra rotoli e ordini.

**int getParetoFrontsCount()** restituisce il numero di KQI valutati all'interno della funzione di fitness.

**int getParetoSetCount()** restituisce il numero di geni all'interno di una soluzione.

**int getLinesNumber(int idStep)** ritorna il numero di linee attive per lo step identificato da *idStep*.

**int getNKQIbyStep(int idStep, int idLinea)** restituisce il numero di KQI valutati dalla linea *idLinea* appartenente all'*idStep*.

E' stato necessario implementare le due funzioni *int getParetoFrontsCount()* e *int getParetoSetCount()* in quanto necessarie per il corretto funzionamento delle routine *getParetoFrontElem()* e *getParetoSetElem*. Le funzioni che prelevano le righe dalle matrici non hanno idea di quanti elementi sono contenuti all'interno di una riga delle matrici a cui accedono quando vengono invocate.

### 4.3.2 View

La view non è altro che l'interfaccia operatore tramite cui è possibile inserire i dati in input al simulatore e, una volta terminata la simulazione, di poter ottenere i dati di output. Questo aspetto che verrà approfondito nel capitolo 5

## 4.4 Design Pattern Strategy

Lo strategy pattern è uno schema architetturale comportamentale<sup>3</sup>, l'obiettivo di questa architettura è encapsulare un algoritmo all'interno di un oggetto: è utile in quelle situazioni dove sia necessario modificare dinamicamente gli algoritmi utilizzati da un'applicazione. Prevede infatti che gli algoritmi siano interscambiabili tra loro in modo trasparente al software che ne fa uso. La famiglia di algoritmi che implementa una certa funzione esporta sempre la medesima interfaccia, in questo modo il client dell'algoritmo non deve fare nessuna assunzione su quale sia la strategia istanziata in un particolare istante. Una tipica implementazione del pattern è la seguente:

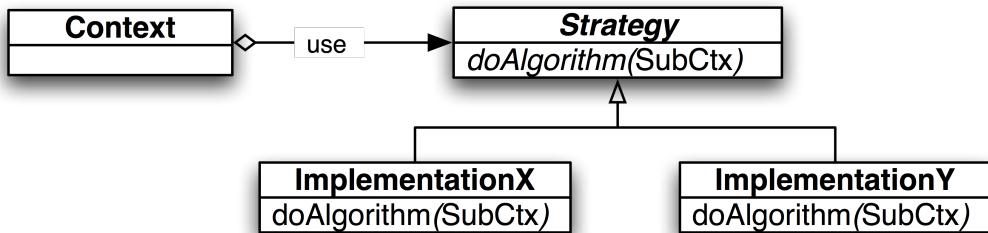


Figura 4.4: Esempio di struttura dello strategy pattern

Nel caso del tirocinio, il Context (contesto) in figura corrisponde alla classe Plant la quale usa la classe Strategy (equivalente alla classe GAStrategy del progetto). Implementation X e Implementation Y non sono altro che i tipi di GA usati i quali mettono a disposizione la funzione RunGA(); lo

<sup>3</sup>[http://it.wikipedia.org/wiki/Strategy\\_pattern](http://it.wikipedia.org/wiki/Strategy_pattern)

strategy pattern è stato impiegato per poter utilizzare gli algoritmi genetici descritti al termine del paragrafo 3.1.

## 4.5 Implementazione funzione di fitness

Nel paragrafo 3.7 è stato descritto il compito della funzione di fitness, passiamo ora a spiegare nel dettaglio le scelte effettuate per la sua implementazione, cominciando da questo estratto di codice:

Funzione di fitness

```
DMatrix Plant::fitness(DMatrix& population) {
    /* per ogni soluzione */
    for (int i = 0; i < popRows; i++) {
        /* matrice di valutazione della popolazione in←
           ingresso */
        DMatrix fit(popRows, N_KQI, DBL_MAX);
        /* workload medio */
        double WL_average = 0;
        /* converto soluzione in cromosoma */
        DNASequencer seq(population.getRow(i));
        /* per ogni individuo valuto i KQI di Crossbow←
           e Flatness (ottimizzazione) */
        DVector CBFL = PLANT.findMatches(seq);
        /* per ogni step dell impianto */
        for (int step = 0; step < PlantSteps; step++) {
            /* vettore di WL dello step, un elemento ←
               per ogni linea */
            DVector wl;
            /* per ogni linea dello step */
            for (int line = 0; line < PLANT.←
                GetProcessingStep((ProcessingStepType)←
                step)->GetLineCount(); line++) {
```

```

        /* calcolo valore del WL per ogni ←
           linea */
        DVector v = PLANT.GetProcessingStep((←
            ProcessingStepType)step)->GetLine(←
            line)->KQIEvaluation(seq);
        /* aggiungo calcolo del WL */
        wl.add(v[ID_WL]);
    }
    /* prendo il valore massimo per quello ←
       step */
    WL_average += wl.max_val();
}
/* calcolo media WL */
WL_average /= (double) PlantSteps;
/* contiene valore dei KQI relativi alla ←
   soluzione */
DVector val(0);
/* aggiungo media WL */
val.add(WL_average);
/* aggiungo min CB */
val.add(CBFL.getValueAt(ID_CB));
/* aggiungo min FL */
val.add(CBFL.getValueAt(ID_FL));
/* inserisco valori relativi al cromosoma ←
   nella matrice di fitness */
fit[i] = val;
}
/* ritorna al GA il vettore contenente le ←
   valutazioni effettuate per ogni soluzione */
return fit;
}

```

La funzione di fitness prende in ingresso una popolazione (insieme) di soluzioni derivate attraverso l'applicazione delle funzioni di crossover e mutazione. Nel nostro caso essa valuta uno per uno i planning proposti ed

effettua la chiamata dei modelli di KQI attraverso:

**findMatches(DNASequencer seq)** questa funzione si occupa di valutare la soluzione *seq* in base ai valori predetti dai modelli di KQI in termini di crossbow e flatness; si occupa quindi di valutare i KQI globali. Inoltre una volta fatto ciò, si occupa anche di determinare gli accoppiamenti tra rotoli e ordini. La routine verrà approfondita nel successivo paragrafo.

**KQIEvaluation(DNASequencer seq)** tramite questa funzione è possibile valutare i KQI di linea come il workload. Il calcolo del workload medio viene effettuato all'interno della funzione di fitness, il valore massimo preso tra tutte le linee appartenenti ad ogni processo partecipa al calcolo del workload medio della soluzione.

A questo punto tutte le soluzioni appartenenti alla popolazione generata nei passi precedenti del GA è stata valutata. Se è stato raggiunto il numero massimo di iterazioni consentito, oppure le soluzioni riescono ad azzerare lo scostamento di crossbow e flatness e contemporaneamente, a ottenere un carico di lavoro equidistribuito tra tutti i processi, l'algoritmo termina. Altrimenti le soluzioni proposte vengono utilizzate per eseguire un ulteriore iterazione del GA e l'algoritmo continua la sua esecuzione alla ricerca di soluzioni migliori.

È importante sottolineare che le valutazioni effettuate tendono a ridurre al minimo casi di overquality dei rotoli per garantire una qualità sufficiente alle richieste ma che non aumenti i costi di produzione.

## 4.6 Ottimizzazione

Nel corso del tirocinio all'interno del software sono stati integrati algoritmi di ottimizzazione che sfruttano i KQI predefiniti come obiettivi, e calcolano una pianificazione ottimale della sequenza di lavorazioni sulla base degli ordini dei clienti, come introdotto nel paragrafo 2.4.

Questo problema di far sì che sia crossbow che flatness vengano minimizzate per effettuare l'associazione tra ordini e rotoli, a prima vista può

essere ridotto ad un ulteriore problema di tipo multi-obiettivo, più precisamente a un MOAP (Multi Objective Assignment Problem) ossia un problema multi-obiettivo di assegnamento.

Dopo aver valutato possibili alternative siamo giunti alla conclusione che trasformare un problema multi-obiettivo in un problema con un'unica funzione obiettivo da minimizzare sarebbe stata la via migliore da seguire (oltre che la meno complessa). Tramite l'utilizzo della teoria dei grafi è stato possibile trasformare il problema multi-obiettivo in un problema di accoppiamento MBMP (Maximum Bipartite Matching Problem, Accoppiamento Bipartito con Cardinalità Massima).

Un accoppiamento in un grafico bipartito<sup>4</sup> è un insieme di archi scelti in modo tale che nessuno degli archi condivide un estremo. Un accoppiamento massimale è un accoppiamento di dimensione massima (numero massimo di archi), ovvero dove ogni nodo è collegato esclusivamente ad un altro nodo il quale non è collegato ad un ulteriore nodo. Presa una soluzione si definiscono due insiemi di nodi che rappresentano uno l'insieme dei rotoli lavorati (in base alle previsioni dei KQI), e l'altro che rappresenta l'insieme degli ordini; a questo punto si va ad analizzare se, in base alle predizioni uno o più ordini vengono soddisfatti. Per ogni rotolo se un'ordine viene soddisfatto viene tracciato un arco che collega il nodo *id\_rotolo* al nodo *id\_ordine* che soddisfa i vincoli, il peso di tale arco è sempre di costo unitario.

Dopo la valutazione si giunge a una situazione simile a quella espressa in figura 4.5 in cui le due partizioni sono visivamente separate, ciascun vertice a sinistra è collegato solo a vertici a destra. Un nodo può soddisfare esattamente un ordine come succede per il rotolo 3, più di un'ordine come nel caso dei rotoli 0, 1 e 4, oppure nessun ordine come per il rotolo 2. Adesso abbiamo costruito un grafo bipartito ovvero un grafo non orientato (senza una direzione specifica) e aciclico cioè che non è possibile costruire all'interno del grafo un cammino che presenta un ciclo all'interno.

---

<sup>4</sup><http://www.geeksforgeeks.org/maximum-bipartite-matching/>

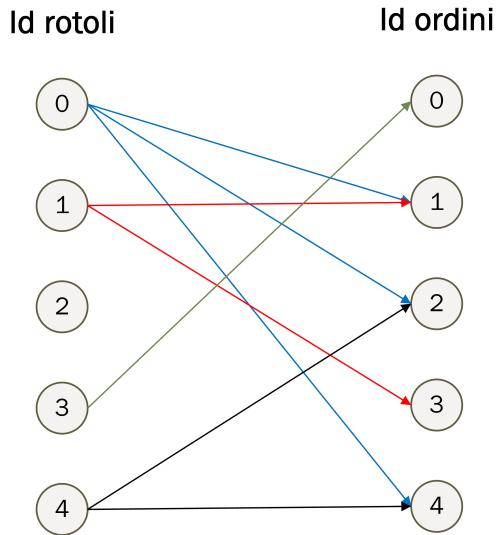


Figura 4.5: Esempio di assegnamento tra rotoli e ordini in un MBMP

La funzione che determina gli accoppiamenti tra rotoli e ordini, minimizzando la differenza tra i valori predetti dai KQI e i valori obiettivo presenti nella lista degli ordini, è stato implementata utilizzando l'algoritmo di Hopcroft-Karp. La complessità di questo algoritmo è nell'ordine di  $O(\sqrt{N}M)$ , dove  $N$  rappresenta il numero complessivo dei nodi che compongono il grafo e  $M$  corrisponde al numero totali degli archi che collegano i nodi tra loro. Con l'approccio che si basa sull'algoritmo veloce della moltiplicazione di matrici la complessità diventa  $O(V^e)$ , indipendente quindi dal numero di archi. Così facendo per una quantità bassa di rotoli come quella per cui verrà utilizzato il software, ossia circa 20-30 rotoli, sarà possibile tollerare un gran numero di possibili accoppiamenti ad ogni iterazione dell'algoritmo; tutto questo snellendo il carico computazionale e quindi le risorse di macchina richieste.

Essa prende in ingresso il grafo (implementato come una matrice di adiacenza) e determina l'insieme degli accoppiamenti in cui la differenza tra i KQI target e quelli predetti viene minimizzato per ogni rotolo. Per fare ciò è stato necessario creare una routine all'interno della funzione fitness, in grado di valutare ogni soluzione proposta dal GA.

La realizzazione dei modelli di KQI non ha fatto parte di questo tirocinio, il loro sviluppo avrebbe dovuto concludersi entro marzo da parte di un altro partner del progetto, ma a causa di problemi non dipendenti dal SIAD, i modelli non sono stati ricevuti in tempo per la scrittura di questa relazione.

È stato quindi deciso di affidarsi a valori generati casualmente all'interno di un range di valori compresi tra zero e uno in maniera completamente stocastica e indipendente da altri fattori. La gamma di valori assunti dai KQI corrisponde a un indice, quando il valore assunto è pari a 0 significa che il pezzo è completamente esente da imperfezioni, mentre quando l'indice assume valore 1 il pezzo risulta completamente difettoso.

In questa maniera è stato possibile sopperire alla mancanza dei modelli ed è stato possibile essere in grado di valutare la correttezza dell'implementazione effettuata anche in mancanza di valori realistici. In pratica tutti questi valori sono stati inseriti all'interno di un file di testo in formato CSV e, per ogni possibile cammino che ogni rotolo può compiere all'interno dell'area di laminazione a freddo, sono stati generati valori per i KQI in uscita dalla linea di rifinitura. Quando all'interno del GA viene generato un macrogene corrispondente a una possibile rottura, la rottura è stata valutata accedendo alla struttura dati contenente il file CSV, prelevando i valori dei KQI generati precedentemente.

```

coil_id;route;CB;FI
1;0000;0.620;0.600
1;0001;0.173;0.090
1;0002;0.255;0.859
1;0010;0.911;0.700
1;0011;0.725;0.230
1;0012;0.576;0.811
1;0020;0.404;0.988
1;0021;0.090;0.321
1;0022;0.511;0.061
1;0100;0.726;0.557
.
.
.
2;0000;0.377;0.735
2;0001;0.954;0.543
2;0002;0.540;0.311
2;0010;0.071;0.182
2;0011;0.093;0.463
2;0012;0.009;0.915
2;0020;0.643;0.001
2;0021;0.030;0.208
2;0022;0.455;0.127
2;0100;0.009;0.727
2;0101;0.354;0.780
2;0102;0.437;0.437
.
.
.
3;0000;0.553;0.958

```

Figura 4.6: Contenuto del file che racchiude i valori dei KQI generati casualmente per ogni possibile rotta di un rotolo

Come si evince dalla figura 4.6 la struttura del file è così definita:

**Id Coil** identificatore univoco del rotolo.

**Route** la rotta predetta dal simulatore per quel determinato rotolo.

**CB** valore di distorsione a balestra in uscita dalla linea di rifinitura in base alla Route.

**FL** valore di planarità in uscita dalla linea di rifinitura in base alla *Route*.

È giusto ricordare che per quanto riguarda il workload non si è rivelata necessaria la richiesta di una libreria dedicata in quanto questo modello è stato implementato all'interno del software.

L'implementazione di questa importante funzione è riassunta nel seguente estratto di codice:

#### Funzione di matching tra ordini e bobine lavorate

```
DMatrix Plant::findMatches(DNASequencer& x){  
    /* contiene valori di CB e FL per i-esimo coil */  
    DVector CBFLVals;  
    /* contiene valori di CB e FL per tutti i coil */  
    DMatrix GlobalCBFLVals(nCoils, N_GLOBAL_KQI);  
    /* somma totale per CB */  
    double CBSum = 0;  
    /* somma totale per FL */  
    double FLSum = 0;  
    /* matrice restituita contenente la media di CB e ←  
       FL relativa alla soluzione */  
    DVector ret(N_GLOBAL_KQI);  
    /* matrice di adiacenza del grafo bipartito */  
    TMatrix <bool> weight(nCoils, nOrders);  
    /* per ogni coil appartenente al cromosoma */  
    for (int i = 0; i < nCoils; i++){  
        /* prelevo rotta i-esimo coil */  
        string route = x.getRouteByCoil(i);  
        /* calcolo valori di CB e FL per i-esimo coil ←  
           */  
        CBFLVals = PLANT.findKQIRoutes(i, route);  
        /* salvo nella matrice valutazione i-esimo ←  
           coil della soluzione */
```

```

GlobalCBFLVals.getRow(i) = CBFLVals;
/* vedo quali ordini sono soddisfatti dai ←
   valori di lavorazione */
for (int j = 0; j < nOrders; j++){
    /* se sia CB che FL sono sufficienti */
    if (ordersList[j][ID_CB] >= CBFLVals[ID_CB←
        ] && ordersList[j][ID_FL] >= CBFLVals[←
        ID_FL])
        /* il j-esimo ordine viene soddisfatto←
           da i-esimo coil, creo un arco */
        weight[i][j] = true;
    else weight[i][j] = false;
}
/* contiene assegnamenti tra coil e ordini */
IVector MatchR(nCoils, -1);
/* chiamo algoritmo di MBM */
maxBPM(weight, MatchR);
/* i corrisponde a indice coil, il contenuto ←
   rappresenta indice ordine */
for ( int i = 0; i < MatchR.size(); i++){
    /* coil non assegnato, al calcolo dei KQI ←
       sommo un valore di massima penalita */
    if (MatchR[i] == -1){
        CBSum = KQIPenalty(CBSum);
        FLSum = KQIPenalty(FLSum);
        break;
    }
    /* coil assegnato */
    CBSum += (GlobalCBFLVals[i][ID_CB] - ←
              ordersList[MatchR[i]][ID_CB]);
    FLSum += (GlobalCBFLVals[i][ID_FL] - ←
              ordersList[MatchR[i]][ID_FL]);
}
/* inserisco valori medi per CB e FL */

```

```

    ret[ID_CB] = CBSum / nCoils;
    ret[ID_FL] = FLSum / nCoils;
    return ret;
}

```

La funzione di ottimizzazione `findMatches()` si avvale dell'uso di altre routine il cui compito è di:

**bool maxBPM(BMatrix& *bpGraph*, IVector& *matchR*)** implementa l'algoritmo di Hopcroft-Karp,

**string getRouteByCoil(int *coilIdx*)** ritorna la rotta del rotolo *coilIdx* all'interno del cromosoma su cui viene invocato.

**DVector findKQIRoutes(int *lineIdx*, string *route*)** in base all'indice *lineIdx* e alla rotta *route* accede al file dei KQI di figura 4.6 e ritorna un vettore contenente i valori assunti dai KQI al termine della rotta.

Mentre utilizza la struttura dati:

**DMatrix *ordersList*** è una matrice di valori in virgola mobile contenente per ogni ordine i valori di crossbow e flatness target.

## 4.7 Un possibile scenario

Il diagramma di sequenza in figura 4.7 esemplifica un tipico scenario di utilizzazione del software da parte di un operatore.

In particolare viene mostrato il comportamento dell'applicazione sotto il punto di vista dell'ottimizzazione della pianificazione. Il sistema prende in input la lista degli ordini, la lista dei rotoli disponibili e il file di configurazione dell'impianto; tramite quest'ultimo viene costruito dinamicamente l'impianto su cui verrà effettuata la simulazione. Una volta fatto ciò viene inizializzata l'esecuzione del GA che provvede a generare e testare le soluzioni in base ai valori predetti dai modelli di KQI. Questo compito viene iterato fino ad ottenere un'insieme di soluzioni che rispettano i vincoli sui KQI dettati dalla lista degli ordini. Una volta terminata la simulazione, l'operatore può seguire

le indicazioni fornite dal simulatore per programmare il lavoro all'interno dell'area.

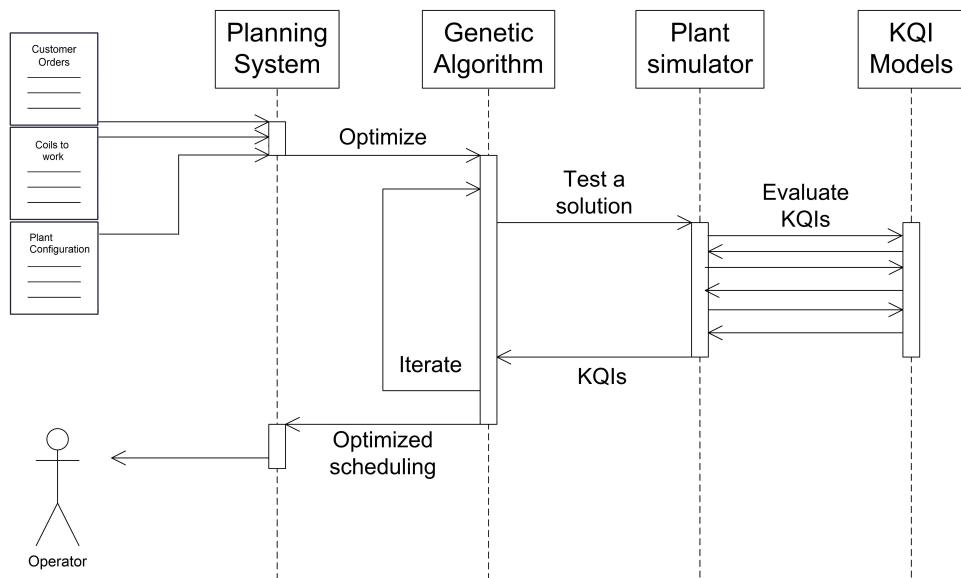


Figura 4.7: Diagramma di sequenza

# Capitolo 5

## Interfaccia grafica

In questo capitolo viene presentata l’interfaccia grafica o GUI (Graphics User Interface, interfaccia grafica utente) realizzata per poter utilizzare la libreria sviluppata nel capitolo 4.

Tramite l’interfaccia grafica è possibile inserire i file di input contenenti gli ordini ricevuti e i rotoli disponibili per la lavorazione ed infine il file di configurazione dell’impianto. Con quest’ultimo file è impostare la dimensione dell’impianto come numero di linee per ogni processo, il numero di KQI da valutare, le librerie da caricare per i modelli di KQI, eccetera. Una volta terminata la simulazione, attraverso l’interfaccia vengono mostrati a schermo il Fronte di Pareto ovvero l’insieme di tutte le possibili soluzioni (gli eventuali planning di produzione per ogni ordine ricevuto) e l’Insieme di Pareto che corrisponde ai valori medi assunti da ogni KQI per ogni soluzione del fronte. Sempre tramite la GUI è possibile esportare i dati di output, così che i risultati possano essere esaminati anche al di fuori dell’ambito del simulatore.

In questo capitolo verrà introdotta la schermata principale con un approfondimento dei vari componenti che la costituiscono ed i legami che essi hanno con i risultati forniti dall’utilizzo dei GA.

## 5.1 Layout

Esiste un solo tipo di schermata per quanto riguarda la GUI la quale è sufficiente all'operatore per inserire i dati di input ed a contenere tutti i dati ottenuti dalla simulazione. Così è come si presenta la schermata principale all'avvio dell'applicazione:

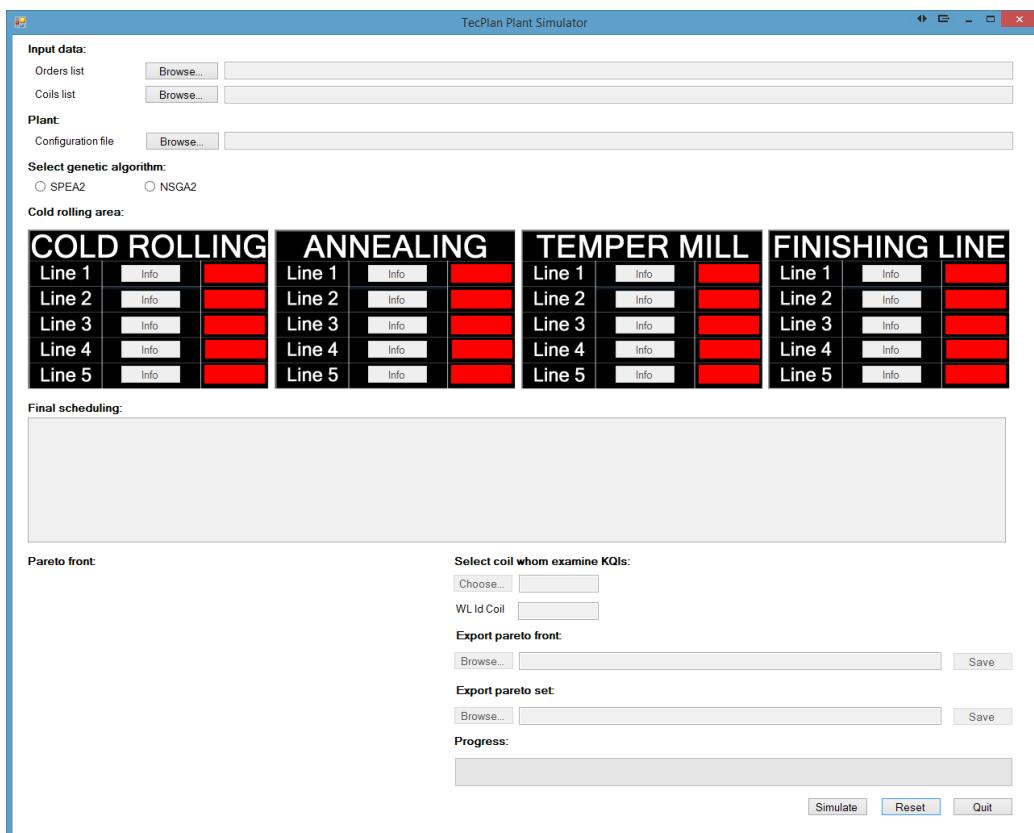


Figura 5.1: Schermata principale prima dell'esecuzione della simulazione

Mentre nella seguente figura viene illustrata la stessa schermata dopo aver eseguito la simulazione:

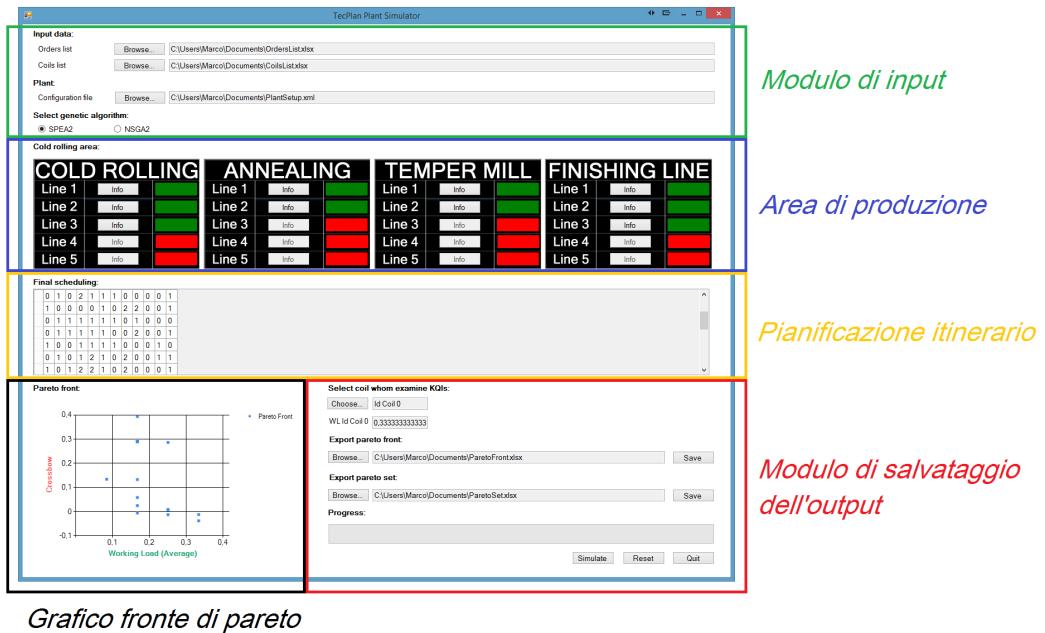


Grafico fronte di pareto

Figura 5.2: Schermata principale dopo l'esecuzione della simulazione

Nella figura 5.2 sono stati evidenziati i cinque form che compongono la schermata:

**Modulo di input (Input data)** modulo tramite cui inserire i dati di input, lista rotoli disponibili, lista ordini ricevuti, file di configurazione, scelta GA.

**Area di produzione (Cold rolling area)** per ogni processo produttivo mostra le linee attive così come imposto dal file di configurazione.

**Pianificazione itinerario bobine (Final scheduling)** mostra per ogni rotolo il percorso da effettuare all'interno dell'area di produzione, ovvero le linee di ciascun processo in cui il rotolo andrà lavorato.

**Grafico fronte di pareto (Pareto chart)** mostra l'insieme di Pareto ovvero l'insieme di soluzioni ottime del problema così come descritto nel paragrafo 3

**Modulo salvataggio output (Data export form)** permette il salvataggio del fronte e dell'insieme di Pareto in formato Microsoft Excel '97.

i loro compiti saranno approfonditi nei successivi paragrafi.

## 5.2 Modulo di input

Tramite il modulo di input è possibile fornire i dati in ingresso al simulatore, i quali comprendono:

- File contenente la lista degli ordini.
- File contenente la lista dei rotoli disponibili.
- File di configurazione dell'impianto.
- Selezione del GA da utilizzare.

È compito dell'operatore compilare il form ed una volta terminato, premere il pulsante Simulate presente nel modulo di salvataggio dell'output per effettuare la simulazione.

Input data:	
Orders list	Browse... C:\Users\Marco\Documents\OrdersList.xlsx
Coils list	Browse... C:\Users\Marco\Documents\CoilsList.xlsx
Plant:	Configuration file Browse... C:\Users\Marco\Documents\PlantSetup.xml
Select genetic algorithm:	
<input checked="" type="radio"/> SPEA2	<input type="radio"/> NSGA2

Figura 5.3: Modulo di input compilato

La lista degli ordini deve contenere i seguenti campi:

**Id Ordine** identificatore univoco dell'ordine.

**Obiettivo planarità** il valore di tolleranza da minimizzare per il KQI di planarità.

**Obiettivo** valore obiettivo da minimizzare per il KQI di balestra.

Mentre invece un tipico file delle bobine disponibili comprende le seguenti informazioni:

**Id Coil** identificatore univoco del rotolo.

**Lunghezza** indica la lunghezza del rotolo in metri.

**Larghezza** denota la larghezza del rotolo in metri.

**Qualità** è un aggettivo che individua il tipo di qualità del rotolo.

ID_COIL	LENGTH	WIDTH	QUALITY
0	3.74	1.00	Good
1	4.90	1.15	Ultra
2	7.00	1.35	Poor

ID_ORDER	FLATNESS_TARGET	CROSSBOW_TARGET
0	0.1	0.05
1	0.3	0.3
2	0.2	0.1

Figura 5.4: Contenuto dei file relativi alla lista delle bobine da lavorare (a sinistra) e agli ordini (a destra)

Entrambi i file sono in formato tabellare secondo lo standard di Microsoft Excel 97.

Per rendere il software riusabile è stato deciso di poter configurare l'impianto secondo le diverse esigenze che possono presentarsi. I parametri dei modelli utilizzati per l'astrazione dell'impianto posso essere impostati a piacimento tramite un file di configurazione in formato XML. Per ognuno dei processi è possibile settare il numero delle linee appartenenti a tale processo, e per ogni linea è possibile decidere quali KQI valutare. È inoltre possibile indicare al simulatore dove si trovano i file contenenti i modelli dei KQI da caricare.

```

<?xml version="1.0" encoding="utf-8"?>
<Tecplan>
  <GeneralInfos>
    <application_name>TecPlan Plant Simulator</application_name>
    <author>SSSA - SIAD, Division of PERC.RO. Laboratory</author>
  </GeneralInfos>
  <PlantConfig>
    <ColdRolling linesNumber="2" KQIS_LINE1="Flatness, Crossbow" KQIS_LINE2="Flatness, Crossbow" />
    <Annealing linesNumber="3" KQIS_LINE1="Flatness" KQIS_LINE2="Crossbow" KQIS_LINE3="Flatness, Crossbow" />
    <TemperMill linesNumber="3" KQIS_LINE1="Flatness, Crossbow" KQIS_LINE2="Crossbow" KQIS_LINE3="Flatness, Crossbow" />
    <FinishingLine linesNumber="2" KQIS_LINE1="Flatness, Crossbow" KQIS_LINE2="Crossbow" />
  </PlantConfig>
  <KQISConfig>
    <FlatnessModel path="c:/kqi_models/flatness.dll" />
    <CrossbowModel path="c:/kqi_models/crossbow.mat" />
  </KQISConfig>
  <WhatElseStuff>...</WhatElseStuff>
</Tecplan>

```

Figura 5.5: Esempio di file di configurazione

### 5.3 Area fasi di produzione

Questo modulo mostra all'operatore lo stato attuale dell'impianto, così come settato tramite il file di configurazione.

Dall'esempio di figura 5.6 si possono notare quattro grandi blocchi uno per ogni processo produttivo. È stato posto il limite di cinque linee per ogni processo, accanto al nome di ogni linea è presente rettangolo che si presenta in due colori; verde se la linea è presente, rossa se non è presente nell'impianto. Nell'esempio si nota come il processo di laminazione a freddo abbia tre linee, quello di ricottura due, esattamente come il treno di rinvenimento, mentre la linea di rifinitura abbia tre linee.

Al termine della simulazione, è possibile cliccare sul tasto Info posto accanto al segnale di linea presente, per poter vedere i valori dei KQI relativi a quella stessa linea.

Cold rolling area:							
COLD ROLLING		ANNEALING		TEMPER MILL		FINISHING LINE	
Line 1	Info	Info	Info	Info	Info	Info	Info
Line 2	Info	Info	Info	Info	Info	Info	Info
Line 3	Info	Info	Info	Info	Info	Info	Info
Line 4	Info	Info	Info	Info	Info	Info	Info
Line 5	Info	Info	Info	Info	Info	Info	Info

Figura 5.6: Possibile configurazione dell'impianto

## 5.4 Pianificazione dell'itinerario

A simulazione terminata questo modulo mostra le venticinque possibili soluzioni al problema della pianificazione. Ogni riga della tabella rappresenta un cromosoma appartenenti al gruppo di soluzioni prodotte dall'esecuzione del GA e concernenti l'insieme di Pareto. Il cromosoma va interpretato come descritto nel paragrafo 3.3.

Final scheduling:									
0	1	0	2	1	1	1	0	0	0
1	0	0	0	0	1	0	2	2	0
0	1	1	1	1	1	1	0	1	0
0	1	1	1	1	1	1	0	0	0
1	0	0	1	1	1	1	0	0	1
0	1	0	1	2	1	0	2	0	0
1	0	1	2	2	1	0	2	0	0

Figura 5.7: Esempio di pianificazione fornita dal GA

Selezionando una soluzione è possibile vedere a schermo tramite i loro id univoci all'interno dei file corrispondenti, la lista degli accoppiamenti tra rotoli e ordini. Questi accoppiamenti sono determinati dall'algoritmo di ottimizzazione all'interno della funzione di fitness utilizzata per i GA.

Matches list	
Coil ID	Order ID
0	2
1	0
2	1

Ok

Figura 5.8: Lista degli accoppiamenti tra rotoli e ordini

Non è detto a priori che nell’insieme di Pareto le soluzioni fornite siano tutte differenti l’una dall’altra; è anche possibile per quel determinato problema sia possibile trovare una soluzione soltanto. Ciò significa non che il GA abbia fallito, ma che la natura del problema posto è molto complessa e non lascia uno spazio delle soluzioni ampio per un numero ragionevole di iterazioni del GA selezionato.

## 5.5 Grafico fronte di pareto

Il grafico rappresenta il fronte di pareto<sup>1</sup> un insieme di soluzioni ottime, costituito da tutti i punti non dominati, cioè da quei punti per i quali non esiste nessun punto che sia migliore contemporaneamente per tutti gli obiettivi considerati nella funzione di ottimizzazione. Un punto può far parte del fronte di Pareto anche se non domina nessuno, poiché l’importante è che non sia dominato da altri punti. Quest’ultima affermazione può sembrare in contrasto con quanto mostrato in figura 5.9, poiché sono presenti dei punti

---

<sup>1</sup>[http://it.wikipedia.org/wiki/Fronte\\_di\\_Pareto#Fronte\\_di\\_Pareto](http://it.wikipedia.org/wiki/Fronte_di_Pareto#Fronte_di_Pareto)

al centro del diagramma che sembrano essere dominati dagli altri punti presenti sulla sinistra. Questa apparenza si spiega in quanto i punti disegnati sono proiezioni a due dimensioni cioè 2D, di punti che in realtà posseggono e necessitano di tre coordinate per essere rappresentati, e che sono quindi 3D. Questa rappresentazione a due dimensioni dà l'impressione che alcuni punti rappresentati sul grafo siano dominati da altri, ma in realtà tutti questi oggetti posseggono la caratteristica di non essere dominati da altri.

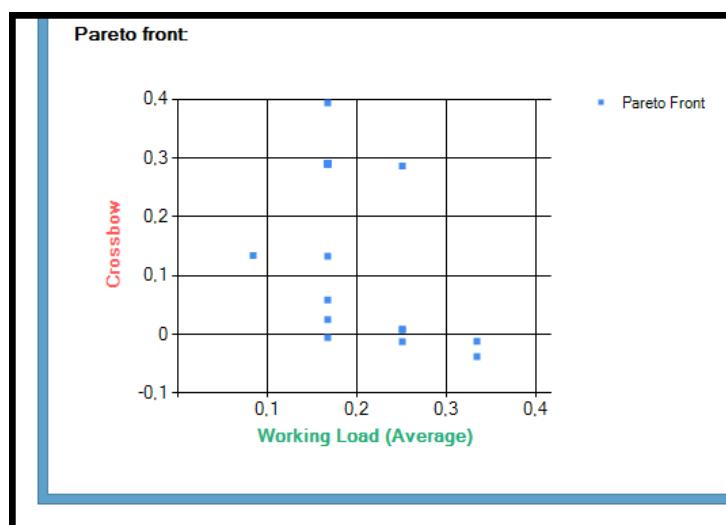


Figura 5.9: Esempi di grafico del fronte di pareto

È inoltre possibile cliccare sulla legenda per poter scegliere i KQI da visualizzare sugli assi del grafo.

## 5.6 Modulo di salvataggio dell'output

Attraverso questo modulo è possibile esportare il fronte di pareto e l'insieme di Pareto secondo il formato Microsoft Excel 97. Ma non solo, una volta selezionata una soluzione dal modulo di pianificazione dell'itinerario, è possibile selezionare una delle bobine appartenenti al cromosoma e visualizzare l'indice di workload per quel rotolo all'interno della soluzione.

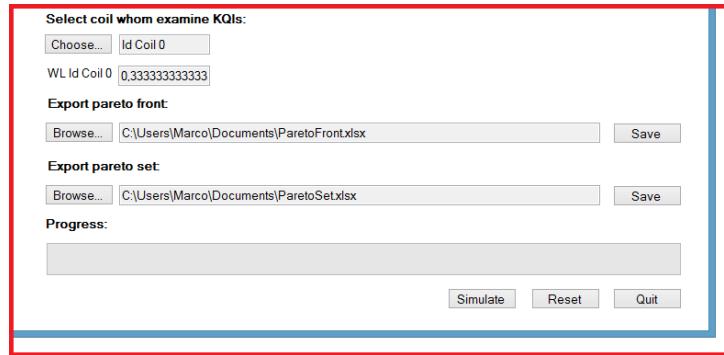


Figura 5.10: Modulo di salvataggio dell'output

Durante la simulazione, la barra di avanzamento mostra la percentuale di lavoro effettuato al fine di ottenere le predizioni del GA utilizzato.

## 5.7 Flusso di controllo

L'algoritmo di esecuzione viene esplicitato tramite il diagramma di flusso, il quale mostra il flusso di controllo dell'applicazione. Come si può notare, vengono effettuati tre tipi di controlli sui file di input, dopo di che si procede con l'esecuzione del GA e vengono mostri a schermo i risultati ottenuti. In realtà viene effettuato un controllo anche sul tipo di GA selezionato, ma è anche possibile che l'operatore non ne selezioni alcuno, a questo punto viene selezionato automaticamente il paradigma SPEA2.

Terminata la simulazione, è possibile rieffettuarla con gli stessi dati in input nel caso in cui le soluzioni proposte non siano di gradimento, oppure esportare le soluzioni proposte.

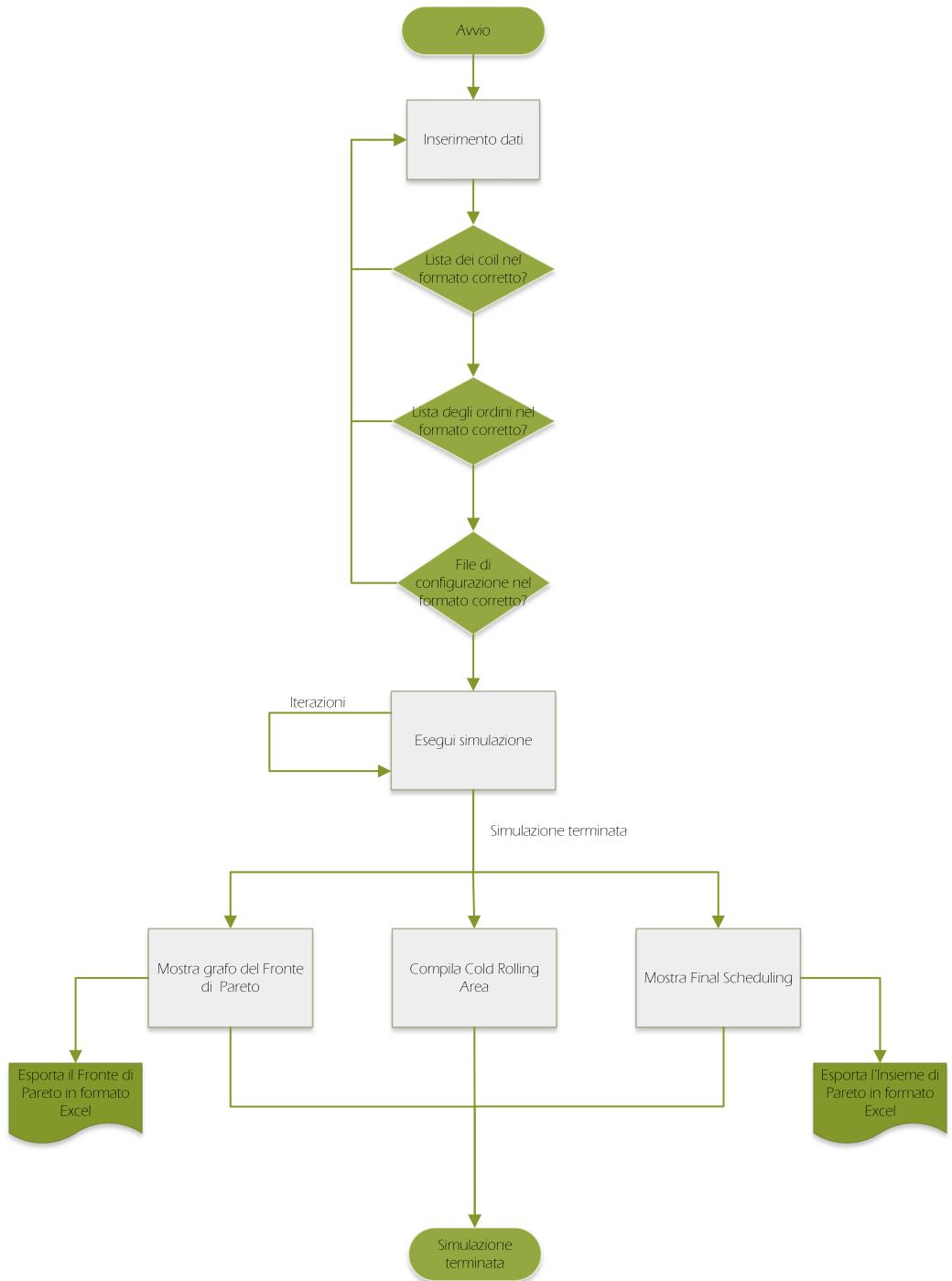


Figura 5.11: Diagramma di flusso relativo alla GUI

# Capitolo 6

## Risultati

Terminata la fase di sviluppo del software sono stati eseguiti test per verificarne il corretto funzionamento, ovvero controllare che le predizioni ottenute sulla pianificazione programmata si avvicinassero il più possibile all'ottimo paretiano. L'ottimo di pareto si realizza quando le funzioni obiettivo sono talmente minimizzate che non è possibile apportare miglioramenti paretiani al sistema, cioè non si può ulteriormente minimizzare il valore di una funzione obiettivo senza aumentare il valore di un'altra. È stato difficile definire delle pietre di paragone per effettuare il confronto tra la soluzione proposta e altri sistemi esistenti. Sul mercato non esiste infatti una tipologia di simulatore che svolge un'ottimizzazione multi-obiettivo come quello del progetto TECPLAN. Vari aspetti che interessano il simulatore sono stati presi singolarmente e utilizzati per lo sviluppo di prodotti commerciali, ma mai tutte le caratteristiche che contraddistinguono TECPLAN, sono state racchiuse all'interno di un solo programma. Inoltre data la natura multi-obiettivo del problema è risultato estremamente difficile creare un set di dati di input per cui a partire da dati valori, si possa prevedere l'output fornito dal simulatore. Selezionare un'insieme di ordini, rotoli e di valori dei KQI che rendesse riproducibile il set di dati in output non è stato realmente possibile data la natura non deterministica del problema. Non è stato quindi possibile procedere con un'analisi quantitativa ma è stato possibile effettuare delle analisi qualitativa.

Sono stati quindi determinati dei parametri variabili:

- Numero dei rotoli presenti per la lavorazione;
- Numero di ordini ricevuti;

Lasciando inalterati i valori per:

- Qualità dei rotoli immagazzinati;
- Qualità desiderata per i rotoli ordinati;
- Previsioni dei KQI selezionati;
- GA utilizzato ovvero SPEA2;
- Numero di passi massimo prima dell'arresto del GA pari a 100.

Per poi di conseguenza valutare la bontà dell'insieme di soluzioni offerte in termini di scostamento dai KQI selezionati. I valori dei KQI per ogni percorso possibile all'interno dell'area di laminazione a freddo, sono stati generati in maniera completamente casuale.

Per quanto riguarda l'impianto è stato deciso di utilizzare sempre la stessa configurazione che è composta da 3 linee per ogni processo di produzione. Ogni linea valuta i KQI relativi a workload, crossbow e flatness.

## 6.1 Creazione del test-set

Per la creazione del test-set è stato determinato il contenuto dei file che includono gli ordini e i rotoli immagazzinati; e sono stati selezionati 9 rotoli e 9 ordini. Partiamo descrivendo il contenuto del file degli ordini:

Possiamo notare fin da subito come esistano in questo insieme ordini con tolleranze elevate come nel caso degli ordini di id 8 e 7, ed ordini che richiedono lavorazioni più accurate come per esempio negli ordini di id 1 e 6. Questo insieme è stato creato per cercare di coprire un panorama il più vasto possibile di range di richieste. Il file contenente i rotoli per il test si compone dei seguenti dati:

Id_Ordine	CB_Target	FL_Target
0	0,45	0,2
1	0,1	0,05
2	0,3	0,3
3	0,4	0,4
5	0,12	0,45
6	0,11	0,12
7	0,3	0,88
8	0,88	0,88

Tabella 6.1: File relativo agli ordini usato per il testing

Id_Rotolo	Length	Width
0	12	7.4
1	15.3	5.2
2	18	3.3
3	9	4.4
4	9.5	7
5	7.6	6
6	22	4.5
7	33	5.6
8	8.1	6

Tabella 6.2: File relativo ai rotoli disponibili usato per il testing

Per quanto riguarda i rotoli immagazzinati invece, i campi relativi a lunghezza e larghezza rotolo non sono ancora dati di input per il simulatore (verranno utilizzati per ulteriori sviluppi), e quindi sono stati scelti casualmente.

Sono stati eseguiti 3 tipi di test, ognuno comprendente:

1. i primi 3 rotoli e i primi 3 ordini;
2. i primi 6 rotoli e i primi 6 ordini;
3. tutti i rotoli e tutti gli ordini disponibili.

## 6.2 Valutazione dei risultati

Le prove effettuate in laboratorio hanno prodotto risultati che hanno centrato in pieno le aspettative. La tabella utilizzata per mostrare i risultati delle prove è composta dalle seguenti colonne:

**Abbinamenti** è una lista composta da coppie del tipo  $(x,y)$  dove  $x$  corrisponde all'Id\_Rotolo e  $y$  corrisponde a Id\_Ordine.

**Workload** coincide con workload medio connesso alla soluzione determinata.

**$\Delta\text{CB}$**  corrisponde all'indice dello scostamento medio del KQI Crossbow per la soluzione determinata.

**$\Delta\text{FL}$**  corrisponde all'indice dello scostamento medio del KQI Flatness per la soluzione determinata.

**Routing** planning previsto per ogni rotolo da codificare secondo quanto stabilito in 3.3.

Per il primo test, ovvero quello con 3 rotoli e 3 ordini sono state prodotte le seguenti possibili pianificazioni:

Abbinamenti	Workload	$\Delta\text{CB}$	$\Delta\text{FL}$	Routing
(0,0) (1,1) (2,2)	0.0833333	0.385667	-0.155333	221121020020
(0,1) (1,2) (0,2)	0.25	0.198667	-0.0963333	221100221100
(0,0) (1,1) (2,2)	0.166667	-0.233333	0.225	120221220021
(0,0) (1,1) (2,2)	0.0833333	0.146667	-0.0636667	221100221000

Tabella 6.3: Dati relativi al primo test, 3 ordini e 3 rotoli

La tabella mostra come con un problema semplice con 3 rotoli e 3 ordini il simulatore sia riuscito ad equidistribuire in maniera apprezzabile il carico di lavoro tra le varie linee. Con 3 linee per ogni processo di produzione, il workload medio dovrebbe corrispondere al 33%, lo scostamento medio dal

carico target è particolarmente basso per la prima, la terza e la quarta soluzione. Lo scostamento dell'indice di crossbow risulta invece mediamente abbastanza alto, ciò significa che se fossero stati utilizzati modelli realistici per i KQI potrebbe essere necessario aggiornare i macchinari di processo. I dati riguardanti lo scostamento di planarità invece risultano più confortanti in eccezione fatta per la terza soluzione. Entrambi i tre tipi di analisi possono pesare sulla decisione finale su quale delle quattro strategie proposte dal simulatore, verrà utilizzata per soddisfare le richieste dei clienti. Gli abbinamenti potrebbero non avere un peso sulla scelta di quale soluzione utilizzare in quanto indicano soltanto le corrispondenze tra rotoli immagazzinati e ordini ricevuti. Passando al secondo tipo di test le cose si complicano: la mancanza di modelli di KQI realistici pesano sul carico computazionale; i valori completamente casuali utilizzati includono anche valori estremamente bassi, i quali inducono il simulatore a trovare meno soluzioni utili di quelle che nella realtà di potrebbero effettivamente trovare. I modelli dovrebbero effettuare predizioni di macchinari costantemente monitorati e tenuti sotto stretta cura, risulterà quindi difficile che i modelli predicono valori bassi al di sotto di una certa soglia, ma valori alti indici di un macchinario in buono stato di manutenzione.

Abbinamenti	Workload	$\Delta CB$	$\Delta FL$
(0,0) (1,1) (2,2) (3,3) (4,4) (5,5)	0	-0.039	-0.039
(0,0) (1,1) (2,2) (3,3) (4,4) (5,5)	0.125	-0.2555	-0.2555
(0,0) (1,1) (2,2) (3,3) (4,4) (5,5)	0.08333333	-0.217333	-0.217333
(0,0) (1,1) (2,2) (3,3) (4,4) (5,5)	0.0416667	-0.205667	-0.205667
Routing			
101101102202112200002221			
101101102202101200002120			
101101102202112200002120			
101101102202112200002220			

Tabella 6.4: Dati relativi al secondo test, 9 rotoli e 9 ordini

Abbinamenti	Workload	$\Delta CB$
(0,8) (1,1) (2,4) (3,2) (4,7) (5,0) (6,6) (7,5) (8,3)	0	-0.196556
(0,8) (1,1) (2,4) (3,2) (4,7) (5,0) (6,6) (7,5) (8,3)	0.0277778	-0.217333
(0,8) (1,1) (2,4) (3,2) (4,7) (5,0) (6,6) (7,5) (8,3)	0.0555556	-0.217556
$\Delta FL$	Routing	
-0.196556	202101101222211210021201211100000220	
-0.217333	202101101222011210021201211100000220	
-0.217556	202101101222101210021201211100000220	

Tabella 6.5: Dati relativi al secondo test, 9 rotoli e 9 ordini

Per i risultati del secondo e terzo test, valgono le considerazioni fatte per la prima prova.

# Capitolo 7

## Conclusioni e prospettive future

Oggetto del tirocinio è stato lo sviluppo di un simulatore della catena di processi di lavorazione di un'azienda siderurgica che predica il planning ottimale in base agli ordini e ai rotoli disponibili.

Il lavoro completamente partito da zero, ha raggiunto gli obiettivi prefissati dalle specifiche iniziali, le quali sono state rispettate nonostante i problemi incontrati e non dipendenti dal datore del tirocinio. Anche se attualmente il simulatore non è stato integrato con i modelli di KQI necessari poiché non ancora pervenuti, la libreria realizzata è completamente indipendente dal formato dei modelli stessi. Sarà possibile così ottenere i modelli in diversi formati a seconda delle necessità del partner produttore e integrarli di conseguenza.

Il sistema creato permette una facile interazione con l'operatore e consente di ottenere in maniera efficiente un'insieme di possibili planning ottimali. Ancora l'affidabilità del sistema consente un abbattimento dei costi di produzione e gestione dei macchinari utilizzati. Attualmente il software è in grado di valutare tre tipi di KQI ma altre tipologie sono state programmate per essere aggiunte in futuro. Inoltre il progetto TECPLAN è destinato ad essere usato per la programmazione della produzione per ogni singolo processo presente nell'area di laminazione a freddo. È prevista un'ulteriore affinamento

del motore di creazione dei cammini; il simulatore dovrà essere in grado di poter prevedere il re-routing dei rotoli, ossia di decidere di tenere conto di ulteriori lavorazioni per i prodotti i cui risultati non risultano quelli desiderati. In maniera automatica dovrà essere in grado di creare rotte più complesse di quelle attualmente generabili con l'ottica di ottimizzare ulteriormente la produzione in termini di macchinari sfruttati e abbattimento dei costi.

# Appendice A

## Acronimi

**BFI** BetriebsForschungsInstitut

**CSV** Comma-Separated Values

**GA** Genetic Algorithm

**GUI** Graphics User Interface

**KQI** Key Quality Indicator

**MBMP** Maximum Biparte Matching Problem

**MOEA** Multi Objective Evolutive Algorithms

**MOAP** MOAP Multi Objective Assignment Problem

**MOP** Multi Objective Optimization Problem

**NPGA** Niched Pareto Genetic Algorithm

**NSGA-II** Non Dominated Sorting Genetic Algorithm II

**PERC.RO.** Perceptual Robotics Laboratory

**RFCS** Research Fund for Coal and Steel

**SIAD** Steel and Industrial Automation Division

**SPEA2** Strength Pareto Evolutionary Algorithm 2

**TECPLAN** TEChnology-based assistance system for production PLAN-ning in stainless steel mills



# Bibliografia

- [1] D.A. Van Veldhuizen C.A. Coello, G.B. Lamont. Evolutionary algorithms for solving multi-objective problems, 2nd ed. 2002.
- [2] K.A. De Jong. Genetic algorithms: A 10 year perspective. pages 169 – 177, 1985.
- [3] L. Thiele E. Zitzler, K. Deb. Comparison of multiobjective evolutionary algorithms: Empirical results. pages 173–195, 2000.
- [4] L. Thiele E. Zitzler, M. Laumanns. Spea2: Improving the strength pareto evolutionary algorithm. 2001.
- [5] N. ; Goldberg D.E. Horn, J. ; Nafpliotis. a niched pareto genetic algorithm for multiobjective optimization. pages 82 – 87, 1994.
- [6] Holland JH. Adaption in natural and artifical system. 1975.
- [7] A. Pratap T. Meyarivan K. Deb, S. Agrawal. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. pages 182 – 197, 2000.
- [8] A. Rong Z. Yang L. Tang, J. Liu. A review of planning and scheduling systems and methods for integrated steel production. pages 1–20, 2001.

# **Ringraziamenti in vernacolo**

## **Pisano**

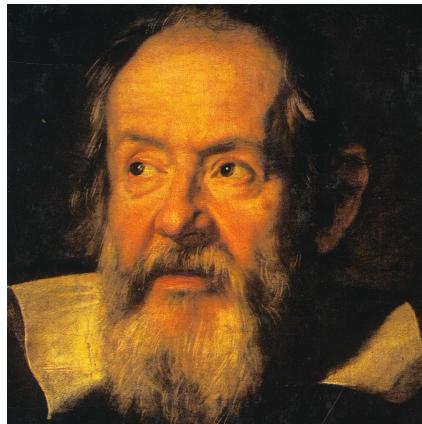
Boia dè finamente ce l'ho fatta, sembrava dovesse un'arrivà mai sto giorno e invece eccomi vì e adesso non si pòle più fa finta di nulla.

L'anni sono passati e anche l'acqua sotto il ponte di mezzo ha fatto lo stesso. Non posso non pensà a chi ha seguitato a stammi dietro in vesti anni, a sopportammi vando mi lamentavo dell'esami andati male o del proff di turno che rompeva in maniera solenne li zebedei (ogni riferimento è puramente casuale). A chi m'escortava vando nei momenti difficili me ne stavo co' capelli ritti in perenne stato d'allerta come i gatti, e intanto passavo il tempo a maledire questo e quello.

Ma soprattutto voglio ringraziare di cuore a chi m'ha dato la forza di reagì con le proprie parole ne peggio momenti, quando tutto sembrava a tinte monocromatiche, e la voglia di mollà tutto era tanta.

Ecco, è proprio avvoi che spero di essere stato un amico e una spalla come voi lo siete stati per me.

E infine ir mi pensiero va a uno dei tanti illustri concittadini che tanta fama hanno portato e continuano a portà nella mi bella città, Galileo Galilei il padre della scienza moderna:



Se un avesse codificato ir metodo scintifio osì come oggi lo conosciamo la scienza cor cavolo che avrebbe potuto fà i passi avanti di cui oggi tutti noi ne godiamo i frutti. Ma soprattutto mi sento di ringraziarLo per i suoi contributi forniti nel campo dell'astronomia di cui io sono diventato appassionato abbestia. Concludo citando una frase tratta dal suo trattato *Il Saggiatore*:

*"La filosofia è scritta in questo grandissimo libro che continuamente ci sta aperto innanzi a gli occhi (io dico l'universo), ma non si può intendere se prima non s'impara a intender la lingua, e conoscer i caratteri, ne' quali è scritto. Egli è scritto in lingua matematica, e i caratteri son triangoli, cerchi, ed altre figure geometriche, senza i quali mezzi è impossibile a intenderne umanamente parola; senza questi è un aggirarsi vanamente per un oscuro laberinto."*

