UNIVERSITÀ DI PISA

Scuola di Ingegneria

Corso di Laurea in Ingegneria Informatica

A.A 2018/19

Titolo Tesi

Valutazione della robustezza delle soluzioni ottime in problemi di scheduling multi-obiettivo delle linee di smontaggio

Candidato

Bill Mono

Relatori

Prof.ssa Beatrice Lazzerini

Dr. Francesco Pistolesi

Indice

1 Introduzione **2**

2 Problema Multi-obiettivo **3**

3 Algoritmi Genetici **4**

3.1 Gene e Cromosoma . . . 4
3.2 Struttura di un GA . . . 5
3.3 Selezione . . . 6
3.4 Operatori di selezione . . . 7
3.4.1 Crossover . . . 7
3.4.2 Mutazione . . . 8
3.5 Nuova generazione . . . 9

4 PlatEMO **10**

5 Svolgimento **10**

5.1 Input del problema . . . 10
5.1.1 Matrice delle precedenze . . . 13
5.1.2 Tempo ciclo . . . 13
5.1.3 Tempi statici e perturbati . . . 14
5.2 Creazione del problema su PlatEMO . . . 14
5.2.1 Funzione Fitness . . . 17
5.3 Soluzioni ottime scelte . . . 17
5.4 Preparazione dei tempi perturbati . . . 18
5.4.1 Esempio di pertubazione dei tempi . . . 18
5.5 Stabilire la probabilità di selezione dei tempi . . . 20
5.5.1 Analisi della probabilità . . . 20
5.6 Analisi della variazione delle soluzioni ottime . . . 23
5.6.1 Valutazione della robustezza . . . 25
5.6.2 Indice Xie-Beni . . . 25

6 Test **26**

7 Conclusioni **30**

8 Ringraziamenti **31**

1 Introduzione

Trovare la miglior sequenza di smontaggio, in un processo di disassemblaggio é uno dei problemi maggiori che le industrie manifatturiere adibite a tale scopo devono cercare di risolvere. Vari metodi nel corso degli anni sono stati proposti nel risolvere tale problema, pur essendo ottimi in presenza di un basso numero di tasks , essi sono risultati inefficaci in presenza di un elevato numero di componenti. Con l'aumentare del loro numero , cresce in maniera esponenziale anche il numero delle possibili combinazioni, il problema di disassemblaggio diventa quindi un problema NP, in questa situazione diventa difficile trovare una soluzione ottima. La soluzione a tale problema é stata trovata facendo ricorso all' utilizzo di algoritmi genetici, i quali riescono a risolvere problemi NP in un tempo generalmente breve. L' obiettivo della mia tesi è quello di ottimizzare il processo di disassemblaggio di un condizionatore utlizzando un algoritmo genetico, cercando di simulare in maniera piú reale possibile, l'esecuzione di un task da parte di un utente. Andando quindi a perturbare i tempi di esecuzione dei vari tasks, in maniera tale da non avere dei tempi statici ma tempi che riflettono l' effettiva esecuzione da parte di un utente, e cercando tra tutte le possibili soluzioni ottime, quella piú robusta. Intendendo per soluzione piú robusta, quella che fra tutte le soluzioni ottime presenta una minor variazione tra il valore ottenuto utilizzando i tempi statici dei tasks componenti tale soluzione, e le soluzioni ottenute perturbando i tempi degli stessi tasks.

2 Problema Multi-obiettivo

Un problema multi-obiettivo è un problema composto da un insieme di obiettivi, rappresentati da un array di funzioni.

$$F(x) = \{f_1(x), f_2(x), f_3(x), f_4(x), f_n(x)\}$$

Un problema multi-obiettivo è generalmente formulato nel modo seguente: .

$$\begin{cases} min & F(x) = (f_1(x), f_2(x), f_3(x), f_4(x), f_n(x))^T \\ x & \\ s.t. & g_i(X) \leq 0, i = 1, ..., p \end{cases}$$

.

Dove $X = (x_1, x_2, x_3, ..., x_D)^T \in \Omega \subset \Re^D$, e rappresenta il *vettore decisione*, mentre Ω è lo *spazio decisione*. $F(x) \in \Lambda \subset \Re^M$ rappresenta invece il *vettore obiettivo*, mentre Λ è lo *spazio obiettivo*, $g_i(X)$ include p differenti constanti.Ogni elemento in X denota una variabile di decisione, e D è il numero di tali variabili. Mentre ogni elemento in F(x) rappresenta un problema di ottimizzazione mono-obiettivo, e M il numero totale di obiettivi. Dati x e y appartenenti all'insieme dei vincoli, si dice che la soluzione x domina y se:

$$f_i(x) \leq f_i(y) \qquad \forall i \in \{1, 2, 3...M\}$$

e

$$\exists j \in \{1, 2, 3, ...M\} \qquad f_j(x) < f_j(y)$$

Nei MOP si parla di ottimizzazione di Pareto. La soluzione x è detto ottimo di Pareto se e solo se $\nexists \;\; y \in \Omega : y \prec x$ L'insieme di tutte le soluzioni ottime di pareto costituisce la curva di Pareto. Le curve di Pareto in molti casi non possono essere calcolate per via analitica, oppure richiedono tempi di risoluzione di ordine esponenziale. Quindi nella risoluzione di questi problemi vengono utilizzati metodi approssimati, gli *algoritmi genetici* i quali anche se non garantiscono di riuscire a trovare la soluzione ottima globale, permettono di trovare buone soluzioni in un tempo limitato.

3 Algoritmi Genetici

Gli Algoritmi Genetici(GA), sono algoritmi di ricerca e ottimizzazione basati su meccanismi dell'evoluzione biologica della specie di Darwin. In natura, gli individui si riproducono mescolando i loro cromosomi, i nuovi individui quindi avranno un patrimonio genetico derivato da entrambi i genitori. La selezione naturale fa si che riescono a sopravvivere e quindi a riprodursi solo gli individui più forti, cioè aventi il valore della funzione da massimizzare più vicino all' ottimo. La fitness media della popolazione tenderà quindi ad aumentare con le generazioni portando cosi la specie ad evolversi nel tempo.

3.1 Gene e Cromosoma

Ogni soluzione del problema da risolvere è rappresentata da una stringa di elementi, *il cromosoma*. Nei problemi di disassemblaggio, il cromosoma identifica la sequenza di decomposizione dell' oggetto. Ogni elemento del cromosoma è detto *gene* , il gene nella catena di disassemblaggio di un oggetto rappresenta il task. .

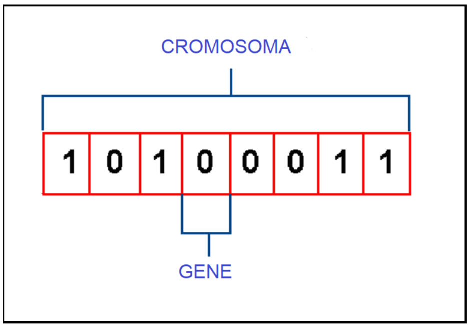


Figura 1: esempio di codifica binaria di un cromosoma, ogni bit della stringa è un gene

3.2 Struttura di un GA

La popolazione è composta da un insieme di individui, essa viene modificata ad ogni generazione attraverso un meccanismo di riproduzione. L' Algoritmo Genetico è quindi caratterizzato da un struttura suddivisa in varie fasi.

1. *Definizione della popolazione iniziale*

   Viene definita anche in maniera casuale un primo insieme di individui, i quali potrebbrero essere possibili soluzioni del problema.

2. *Valutazione di ogni soluzione e selezione delle migliori*

   Viene valutata la qualità di ogni individuo appartenente alla popolazione, tramite la funzione fitness.

3. *Creazione di una nuova generazione*

   La nuova generazione contiene gli individui con qualità più elevata, ma può includere anche altri individui. Tutttavia viene fatto in modo che vi siano una prevalenza di quelli aventi una qualità maggiore.

4. *Condizioni di terminazione*

   Se il numero delle iterazioni è stato raggiunto o se la qualità della migliore delle soluzioni si può considerare accettabile

3.3 Selezione

La selezione di un individuo dipende dal suo valore della fitness, ovvero della qualità di un individuo, un valore più alto di fitness, corrisponde ad una maggiore probabilità di essere scelto come genitore. Grazie al meccanismo della selezione solo gli individui migliori hanno la possibilità di riprodursi e quindi di trasmetttere i loro geni alle generazioni successive. Il meccanismo di scelta più frequentemente utilizzato assegna a ciascun individuo una probabilità di sopravvivenza proporzionale alla sua fitness. Ciò implica che, in media, gli individui più forti si riproducono più frequentemente. Se ad esempio, vi è una popolazione $P(t) = \{x^1, x^2, x^3...x^n\}$, e calcolato il fitess totale

$$F = \sum_{i=0}^{n} f(x^i)$$

.

la probabilità che il cromosoma x^i appartenente a tale popolazione venga scelto è uguale a

$$p_i = f(x^i)/F$$

.

Altri criteri di selezione usati sono:

1. *fitness scaling*

   La fitness di un individuo viene scalata in un modo lineare o per elevazione a potenza secondo coeficienti predefiniti.

2. *ranking* gli individui della popolazione sono ordinati per fitness decrescente ed a ciascuno di loro gli è assegnata una probabilità di produzione in base alla posizione in graduatoria.

3. *tournament selection* si estraggono casualmente n coppie di individui della popolazione e li si confronta fra di loro, si selezionanno per la riproduzione i vincitori di ciascuno confronto.

3.4 Operatori di selezione

Per la creazione di nuovi individui ad ogni generazione si utilizzano due operazioni ispirate dall'evoluzionismo biologico, il crossover e la mutazione.

3.4.1 Crossover

L'operatore di crossover produce un nuovo individuo mescolando l'informazione contenuta in due individui genitori contenuti nella popolazione selezionata.Vi sono vari tipi di crossover, quello che andremo ad utilizzare è l' Order Crossover. L' Order Crossover lavora su coppie di cromosomi, i due punti di taglio sono scelti in maniera casuale. Ciascun figlio è generato prendendo la parte centrale di un genitore e rimpiazzando gli spazi rimanenti con i simboli mancanti presi nell' ordine in cui compaiono nell'altro genitore.

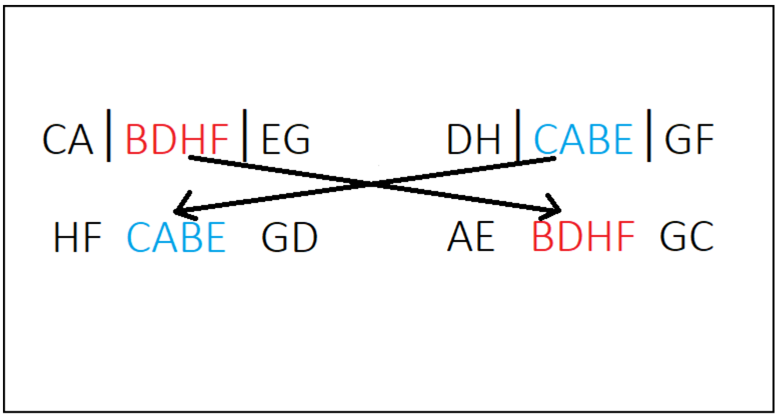


Figura 2: Esempio di Order Crossover

3.4.2 Mutazione

Con la mutazione viene introdotto nuovo materiale genetico in alcuni cromosomi, andando cosi a creare individui con caratteristiche completamente diverse. Data la probabilità di mutazione, per ciascua cifra di un individuo si estrae un numero casuale, se questo numero è maggiore della probabilità di mutazione allora si effettua lo scambio.

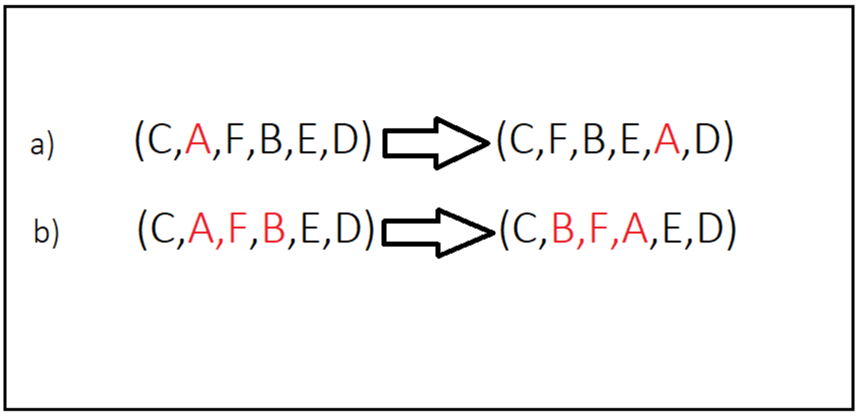


Figura 3: Nella figura a),la mutazione è effettuata togliendo un elemento a caso e reinserendolo in un altra posizione. Nella figura b) viene invece tolta una sottostringa, invertita e reinserita nella stessa posizione.

3.5 Nuova generazione

Ad ogni nuova generazione che si verrà a creare attraverso i metodi di selezione indicati precedentemente andremo incontro ad un miglioramento della fitness, tuttavia bisogna specificare che vi sono vari modi per sostituire la vecchia generazione con la nuova. I due principalmente più utilizzati sono *la selezione steady-state* e *l' ellittismo*. Nella steady-state, ad ogni generazione sono rimpiazzati solamente pochi individui, vengono esclusi dalla nuova generazione gli individui con i cromosomi peggiori, e inseriti al loro posto quelli generati dai migliori. Nell'ellittismo invece si parte dalla considerazione che con le operazioni di croosover e mutazione c'è il rischio di perdere gli individui aventi una qualità superiore, per evitare ciò vengono mantenuti nella generazione successiva gli individui con fitness maggiore, dalla precedente generazione.

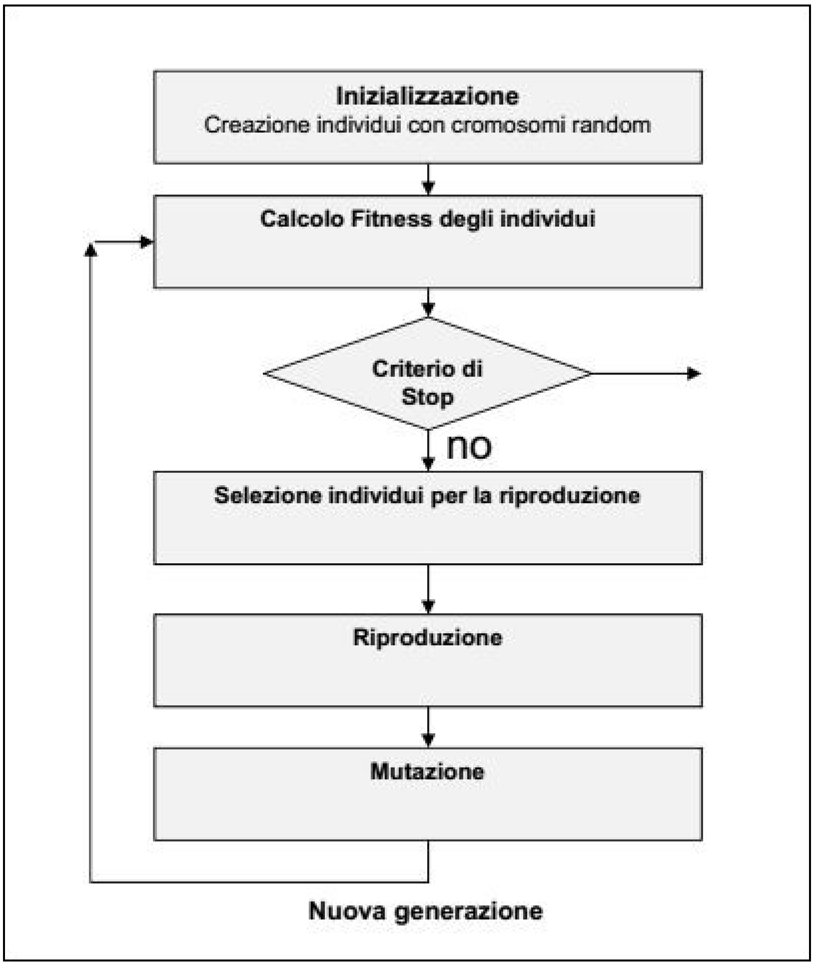


Figura 4: Rappresentazione con uno schema a blocchi dell' algoritmo

4 PlatEMO

La piattaforma che adoperemo per implementare il nostro problema multi-obiettivo è PlatEMO. PlatEMO è un' applicazione basata su MATLAB, utilizzata per ottimizzare problemi multi-obiettivo. Include molti problemi multi-obiettivo, algoritmi evolutivi per la risoluzione di tali problemi, oltre a dei operatori utilizzati in combinazione con gli algoritmi.

5 Svolgimento

Come dichiarato nell' introduzione, il lavoro della mia tesi si è concentrato nel riuscire a ottimizzare il processo di disassemblaggio di un condizionatore, andando ad analizzare gli effetti prodotti dalla variazione dei tempi di esecuzione dei tasks sulle soluzioni ottime , e cercando di trovare la soluzione più robusta tra tutte le possibili soluzioni ottime. Per prima cosa è stato quindi necessario implementare tale problema.

5.1 Input del problema

Il condizionatore preso in considerazione è composto da 64 tasks. Nella creazione del problema è stato innanzitutto necessario creare un file che calcolasse gli input per il condizionatore. In esso vi sono stati salvati il valore del tempo ciclo, la matrice delle precedenze, il costo di manodopera i tempi statici e perturbati di ogni task. Sempre in questo file sono stati calcolati i costi e le caratteristiche dell' oggetto. Di seguito è riportato la vista esplosa dell' oggetto.

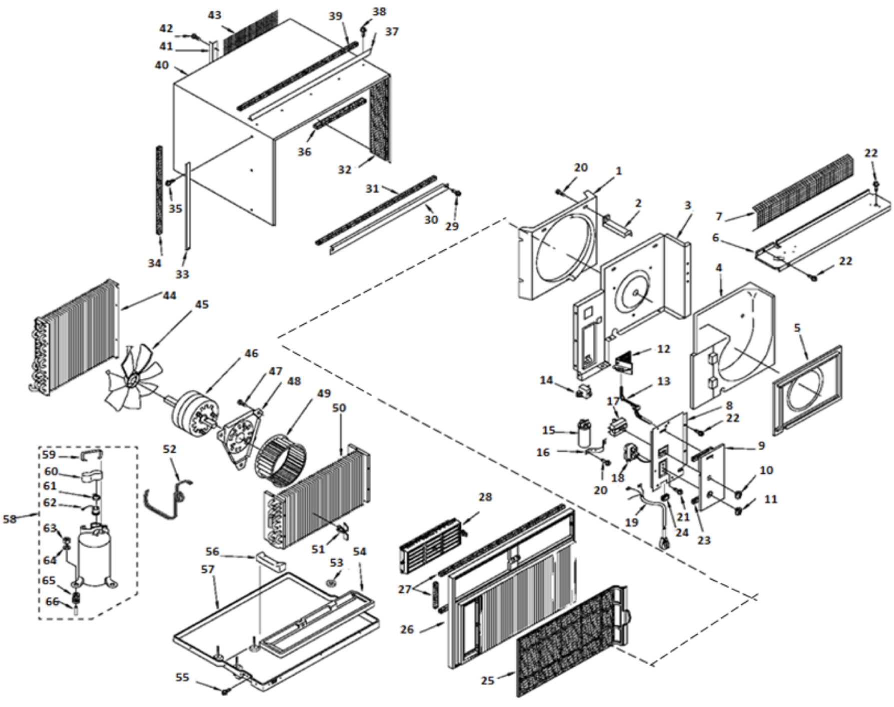


Fig. 7 Exploded view of the air conditioner.

Figura 5: Vista esplosa del condizionatore.

Task i	Precedence constraints	Execution time t_i [min]	Removed parts
1	-	0.90	22
2	-	0.45	20
3	-	0.30	10-11
4	3	0.15	9
5	4	0.30	21
6	4	1.05	23
7	4	0.60	25
8	1,5,6	1.00	8
9	8	0.20	17
10	8	0.30	18
11	2	0.20	16
12	11,13	0.40	15
13	-	0.40	14
14	8	0.80	13
15	14	0.15	12
16	-	1.20	19
17	1	0.60	6-7
18	-	0.30	5
19	18	0.15	4
20	2	0.15	1
21	20	0.20	2
22	12,15,16,19,21	0.15	3
23	-	1.30	42
24	23	0.30	41
25	23	0.15	43
26	-	1.30	38
27	26	0.15	39
28	26	0.30	37
29	-	1.30	35
30	29	0.15	34
31	29	0.30	33
32	-	1.30	29
33	32	0.30	30
34	32	0.15	31
35	23,26,29,32,60	0.25	40
36	35	0.15	36
37	20,33,34	0.40	25
38	37	0.15	26
39	38	0.15	27
40	37	0.15	28
41	35	0.20	51
42	41,50	1.00	50
43	35	1.40	(45,46,47,48, 49)
44	50	1.40	44
45	43	0.20	49
46	43	0.30	47
47	45,46	0.25	48
48	46	0.20	46
49	43	0.10	45
50	35	1.30	52
51	50	1.00	58
52	51	0.15	59
53	52	0.15	60
54	53	0.15	61
55	54	0.15	62
56	51	0.15	63
57	56	0.15	64
58	51	0.15	66
59	58	0.15	65
60	-	1.30	55
61	42,43,44,51	0.15	56
62	42,43,44,51	0.40	53
63	62	0.25	54
64	60,63	0.10	57

Figura 6: Tabella delle corrispondenze tra task eseguiti e pezzi rimossi.

5.1.1 Matrice delle precedenze

La matrice delle precedenze del condizionatore è formata da 64 righe e 64 colonne, è costruita in accordo all'ordine di priorità tra tasks. Se ad esempio abbiamo due tasks i e j, e l'operazione i ha precedenza di esecuzione sull' operazione j, l' elemento in posizione Pij nella matrice avrà valore 1, in caso contrario il suo valore sarà 0. Di seguito è riportata la matrice delle precedenze del condizionatore posto in esame.

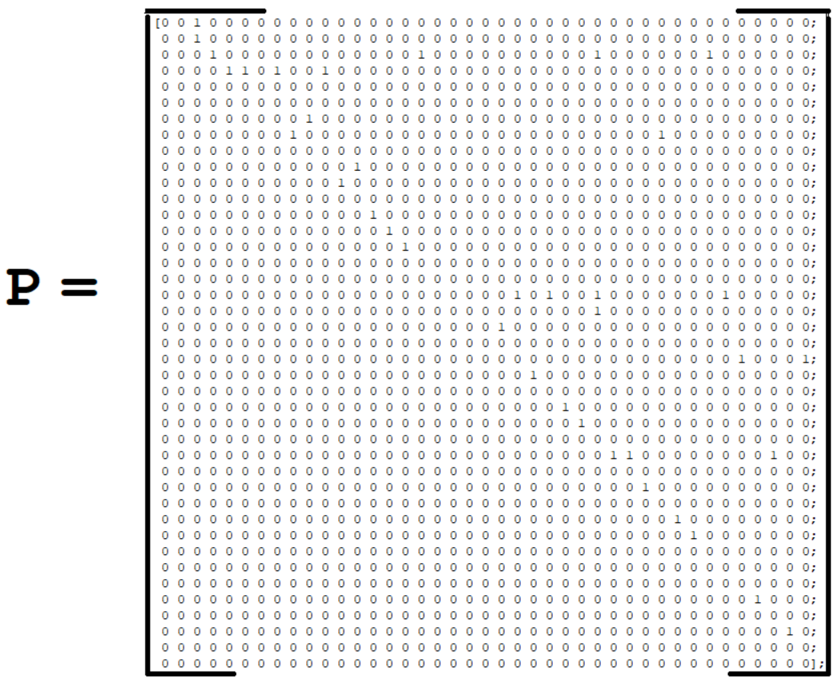


Figura 7: Matrice delle precedenze del climatizzatore.

5.1.2 Tempo ciclo

Per tempo ciclo viene definito il tempo entro il quale un task deve essere eseguito, se un task non viene eseguito entro tale tempo, viene scartato dalla linea di disassemblaggio e si procede con il successivo.

5.1.3 Tempi statici e perturbati

I tempi statici sono i valori medi di esecuzione di ogni task presenti nel vettore dei tempi medi. I tempi perturbati invece sono i tempi di ritardo o di anticipo di esecuzione del task rispetto al suo tempo medio di esecuzione, creati a partire proprio da tale valore.

5.2 Creazione del problema su PlatEMO

Una volta creato il file di input, è necessario creare il problema multi-obiettivo. Questo viene scritto in un file.m su PlatEMO, e posto nella cartella contenente i problemi. L'algoritmo utilizzato per l'ottimizzazione di tale problema è l' NSGAII, mentre l' operatore utilizzato in combinazione con l'algoritmo è l' EApermutation. Di seguito è riportato il codice del problema implementato su PlatEMO:

```

function varargout=DISASSEMBLY_FIX_2(Operation,Global,input)

 global caratteristiche;
    % matrice delle precedenze
    global P;
 % tempo ciclo
    global Tc;
    Tc = 2.18;

    % tempi di smontaggio
    global t;

   global b;

  b = zeros(300,64);
    switch Operation

        case 'init'
            Global.N         = 300;
            Global.N         = 300;
            Global.M         = 3;
            Global.M         = 3;
            Global.D         = 64;
            Global.operator  = @EApermutation;
            Global.evaluation = 10000;
            nVar             = Global.D;
            nIndiv           = Global.N;

            PopDec  = zeros(input,nVar);

              n_seeds = size(seed,1);
              PopDec(1:n_seeds,:) = seed;
```

```
              for i=n_seeds+1:300
                       PopDec(i,:) = randperm(nVar,nVar);
              end

            varargout = {PopDec};

        case 'value'

            nVar              = Global.D;
            nIndiv            = Global.N;

            PopDec = input;

            for i=1:size(PopDec,1)

                fitness = fitness_disassembly(PopDec(i,:));

                PopObj(i,1) = fitness(1);
                PopObj(i,2) = fitness(2);
                PopObj(i,3) = fitness(3);
                b(i,:) = PopDec(i,:);
            end

               save b.mat b;

             PopCon = [];

            varargout = {input,PopObj,PopCon};

        case 'PF'

            nVar              = Global.D;
            nIndiv            = Global.N;

            RefPoint = [0 , 1000  , 1000];
            disp('*************************************');

            varargout = {RefPoint};

        case 'draw'
            nVar              = Global.D;
            nIndiv            = Global.N;

            PopDec = input;

            for i=1:size(PopDec,1)

                fitness = fitness_disassembly(PopDec(i,:));
```

```
                PopObj(i,1) = fitness(1);
                PopObj(i,2) = fitness(2);
                PopObj(i,3) = fitness(3);

            end

            disp('*************************************');

            xlabel('Feasibility');
            ylabel('Profit');
            zlabel('Smoothness');
    end
end
```

.

5.2.1 Funzione Fitness

Nella creazione del problema sono stati posti 3 obbiettivi da ottimizzare

1. *Feasibility*
2. *Profit*
3. *Smoothness*

La funzione fitness utilizzata per valutare la qualità di una sequenza è la $fitness\_disassembly$, essa prende in ingresso il vettore contenente la sequenza e restituisce un vettore contenente il valore dei 3 obiettivi.

5.3 Soluzioni ottime scelte

Una volta messo in esecuzione il problema su PlatEMO, ogni sequenza appartenente alla popolazione iniziale viene valutata tramite la funzione fitness specificata precedentemente. Le sequenze che presentano valori migliori, vengono scelte per la ricombinazione e inserite nella nuova popolazione. Alla fine si otterrà una popolazione contenente tutte le possibili soluzioni ottime. Questa popolazione viene salvata in un matrice e sarà utilizzata successivamente per determinare tra tutte le soluzioni ottime quella più robusta. Abbiamo deciso che per una maggior comprensione, tra tutte le soluzioni ottime presenti sul fronte di Pareto, siano scelte le 5 aventi un profitto più alto. Questa scelta è dovuta al fatto che abbiamo dato un peso maggiore al profitto tra i 3 obbiettivi da ottimizzare. Il grafico sotto riportato, riporta quanto è stato affermato.

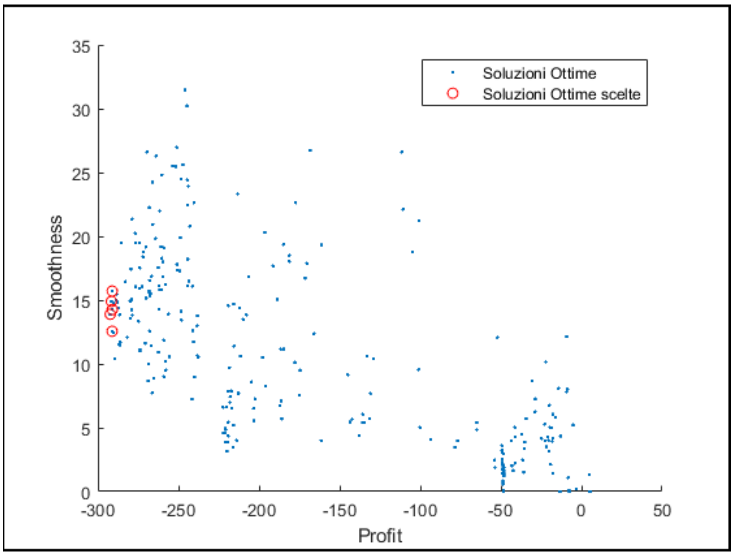


Figura 8: Rappresentazione delle soluzioni ottime e delle soluzioni ottime selezionate.

Il grafico è stato realizzato proiettando il problema avente 3 obbiettivi in uno spazio bidimensionale, in questo spazio non abbiamo considerato la *feasibility* come obiettivo, poichè tale valore è costante e quindi indipendente alle variazioni dei tempi dei tasks. I pallini presenti nel grafico rappresentano le soluzioni ottime ottenute utilizzando il tempo medio di esecuzione di ogni task, quelle cerchiate in rosso sono le 5 soluzioni ottime selezionate.

5.4 Preparazione dei tempi perturbati

Selezionate tra le soluzioni ottime, le 5 soluzioni aventi maggior profitto, dobbiamo ora andare ad analizzare il loro comportamento al variare del valore del tempo dei tasks. Abbiamo già dato la definizione di tempo perturbato, per creare una variazione di tempo prendiamo un elemento dal vettore dei tempi medi, e da esso vengono creati dei tempi di anticipo, quindi di valore minore rispetto al tempo medio e di ritardo , aventi valore maggiore.

5.4.1 Esempio di pertubazione dei tempi

Facciamo un esempio per rendere più chiaro il concetto espresso. Prendiamo dal vettore dei tempi medi $\overrightarrow{t}$, il tempo del task1, l' azione svolta in questo task è lo svitamento di una vite. Il suo tempo medio preso dal vettore $\overrightarrow{t}$ è 0.9, quindi a partire da tal valore quello che è stato fatto è stato quello di variare tale tempo andando a selezionare tempi maggiori e minori rispetto tale valore. La figura sottostante illustra quanto detto. .

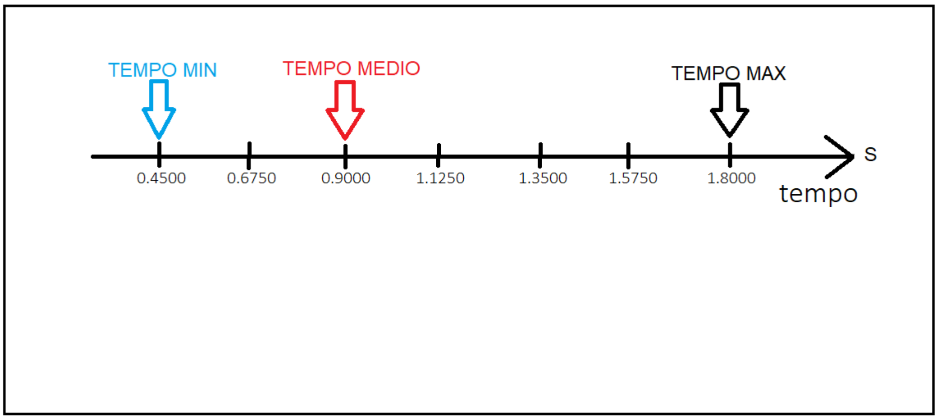


Figura 9: Rappresentazione del tempo medio e dei tempi perturbati.

Nella figura è possibile vedere che oltre al tempo medio, sono stati scelti altri 6 tempi di esecuzione diversi, 2 tempi di valore inferiore rispetto tale tempo, e altri 4 di valore superiore. La scelta di prendere solo 7 istanti, è dovuta al fatto che con 7 istanti di tempo per ogni task vi sono 7 possibili combinazioni, dato che il numero totale di tutti task è 64, in totale avremo quindi $64 * 7 = 448$ possibili vettori perturbati. Un numero non enorme, ma che può diventare tale se al posto di 7 avessimo scelto 100 o 1000 istanti di tempo. La rappresentazione su un eventuale grafico di tutte le possibili soluzioni ottenute utilizzando i vettori dei tempi perturbati sarebbe stata di non facile comprensione. Quindi solamente per una più chiara rappresentazione, è stato scelto questo numero. Comunque sia è bene dire fin da ora che il concetto espresso con 7 istanti di tempo è valido per qualsiasi numero di istanti che si vanno a scegliere. Sotto è riportato il codice MATLAB per ottenere il vettore dei ritardi $\overrightarrow{w}$ a partire da $\overrightarrow{t}$.

```

load('componenticondizionatore.mat');

a = zeros(66,9);
for j=1:66
      a(j,:) = linspace(0,t(j)*2,9);

 end
w = zeros(1,66);

for i=1:66
    w(i)= a(i,randi([3,9]));
end
```

5.5 Stabilire la probabilità di selezione dei tempi

Torniamo adesso ai 7 istanti di esecuzione dei task1, la cui azione comprendeva lo svitamento di una vite. E' ragionevole pensare che questa operazione venga effettuata in un lasso di tempo molto breve.Quindi la probabilità che questa operazione venga eseguita in un tempo breve, deve essere maggiore rispetto alla probabilità che questa sia svolta in un lungo lasso di tempo. Per stabilire la probabilità di selezione e di successivo inserimento nel vettore dei tempi perturbati di un istante di tempo, ho deciso di dividere i componenti del condizionatore in 3 classi, divisi in base alla difficoltà di rimozione di tali parti. Chiaramente si tratta di un giudizio strettamente soggettivo. Nella figura sottostante è presente la tabella con indicato il livello di difficltà di rimozione di ogni pezzo. .

1	Media	21	Bassa	41	Bassa	61	Bassa
2	Media	22	Bassa	42	Bassa	62	Bassa
3	Media	23	Media	43	Bassa	63	Bassa
4	Media	24	Bassa	44	Elevata	64	Bassa
5	Media	25	Elevata	45	Elevata	65	Bassa
6	Media	26	Elevata	46	Media	66	Bassa
7	Media	27	Elevata	47	Media		
8	Bassa	28	Media	48	Media		
9	Bassa	29	Media	49	Elevata		
10	Bassa	30	Bassa	50	Media		
11	Media	31	Bassa	51	Media		
12	Media	32	Bassa	52	Media		
13	Media	33	Media	53	Media		
14	Media	34	Elevata	54	Media		
15	Media	35	Elevata	55	Bassa		
16	Media	36	Elevata	56	Media		
17	Media	37	Media	57	Media		
18	Bassa	38	Bassa	58	Bassa		
19	Bassa	39	Bassa	59	Bassa		
20	Bassa	40	Bassa	60	Bassa		

Figura 10: Divisione delle componenti dell' oggetto nelle 3 classi di difficoltà: Bassa, Media, Elevata.

Per quanto riguarda la probabilità di selezione di un istante, questa come precedentemente detto dipende dalla classe di difficoltà di appartenenza, tuttavia ho deciso che per ogni livello, il numero di tutti i casi possibili rimanga costante in tutte le classi, quello che varia è la redistribuzione delle varie probabilità tra i vari istanti.

5.5.1 Analisi della probabilità

Nella tabella la rimozione di una vite è individuata come operazione semplice da eseguire. Avendo scelto 27 come il numero tutti i casi possibili, la probabilità che l'istante di tempo inserito nel vettore dei ritardi sia il tempo medio di esecuzione del task, è 14/27. Di seguente è riportato il codice MATLAB per ottenere il vettore dei ritardi $\overrightarrow{w}$ a partire da $\overrightarrow{t}$, tenendo conto delle considerazioni fatte sulla probabilità.

```

bassa = [ 2 8 9 10 11 12 14 16 17 18 19 20 21
           22 24 31 32 38 40 41 42 43 55 58 59 60
           61 62 63 64 65 66];


media =   [1 3 4 5 6 7 13 15 23 28 29 30 33
              37 46 47 48  50 51 52 53 54 56 57];

elevata = [25 26 27 34 35 36 39 44 45 49];

prob = zeros(66,27);

for j = 1:length(bassa)

for i = 1:14
prob(bassa(j),i) = a(bassa(j),5);
end

for i = 15:17
prob(bassa(j),i) = a(bassa(j),6);
end

for i = 19:20
prob(bassa(j),i) = a(bassa(j),7);
end

for i = 22:23
prob(bassa(j),i) = a(bassa(j),8);
end

prob(bassa(j),24) = a(bassa(j),9);
prob(bassa(j),25) = a(bassa(j),3);
prob(bassa(j),21) = a(bassa(j),3);
for i = 26:27
prob(bassa(j),i) = a(bassa(j),4);
end

prob(bassa(j),18) = a(bassa(j),4);

end

for j = 1:length(media)

```

```
for i = 1:14
prob(media(j),i) = a(media(j),5);
end

for i = 15:18
prob(media(j),i) = a(media(j),6);
end

for i = 19:21
prob(media(j),i) = a(media(j),7);
end

for i = 22:23
prob(media(j),i) = a(media(j),8);
end

prob(media(j),24) = a(media(j),9);
prob(media(j),25) = a(media(j),3);

for i = 26:27
prob(media(j),i) = a(media(j),4);
end

end

for j = 1:length(elevata)

for i = 1:13
prob(elevata(j),i) = a(elevata(j),5);
end

for i = 15:18
prob(elevata(j),i) = a(elevata(j),6);
end
prob(elevata(j),14) = a(elevata(j),6);
for i = 19:21
prob(elevata(j),i) = a(elevata(j),7);
end
prob(elevata(j),27) = a(elevata(j),7);

for i = 22:23
prob(elevata(j),i) = a(elevata(j),8);
end

prob(elevata(j),24) = a(elevata(j),9);
prob(elevata(j),25) = a(elevata(j),3);


prob(elevata(j),26) = a(elevata(j),4);
```

```
end

for i=1:66
    w(i)= prob(i,randi([1,27]));
end
```

La Figura 10 è una variazione della 9,e mostra gli istanti di esecuzione del task1, dopo le considerazioni fatte sulla probabilità, nello specifico per ogni istante di tempo del task1 è specificato la probabilità di essere selezionato.

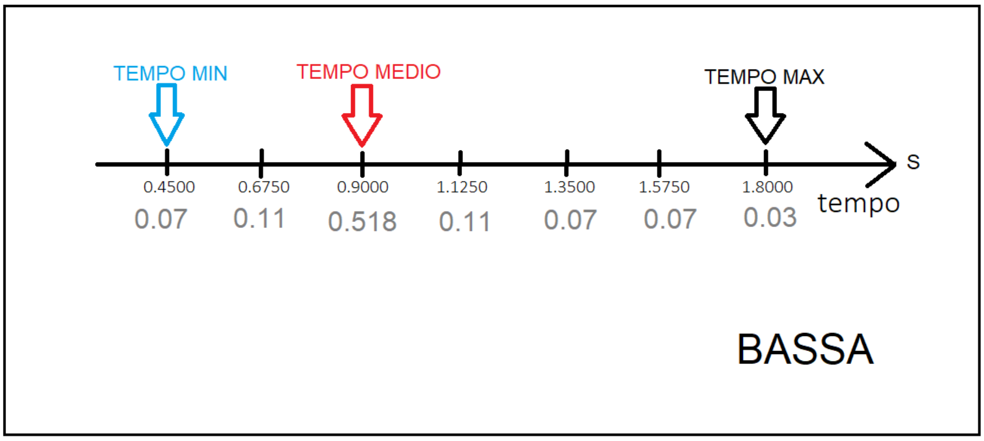


Figura 11: Rappresentazione del tempo medio e dei tempi perturbati con relativa probabilità di essere selezionati.

5.6 Analisi della variazione delle soluzioni ottime

Una volta creati i vettori dei ritardi per ogni soluzione ottima selezionata, possiamo andare ad analizzare la variazione provocata sulle soluzioni da tali vettori.La figura sottostante riporta quanto detto.

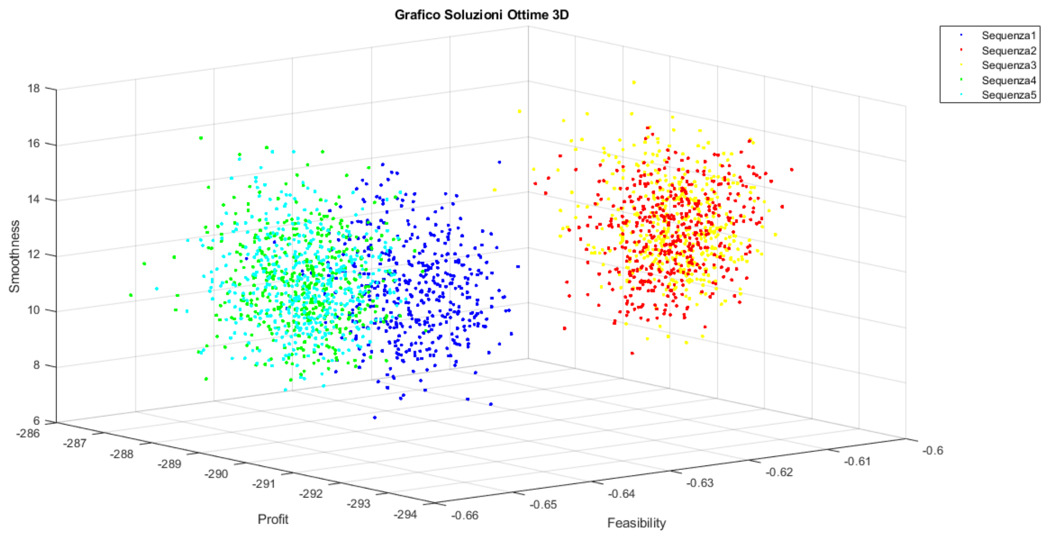


Figura 12: Grafico soluzioni ottime, vista in 3 dimensioni.

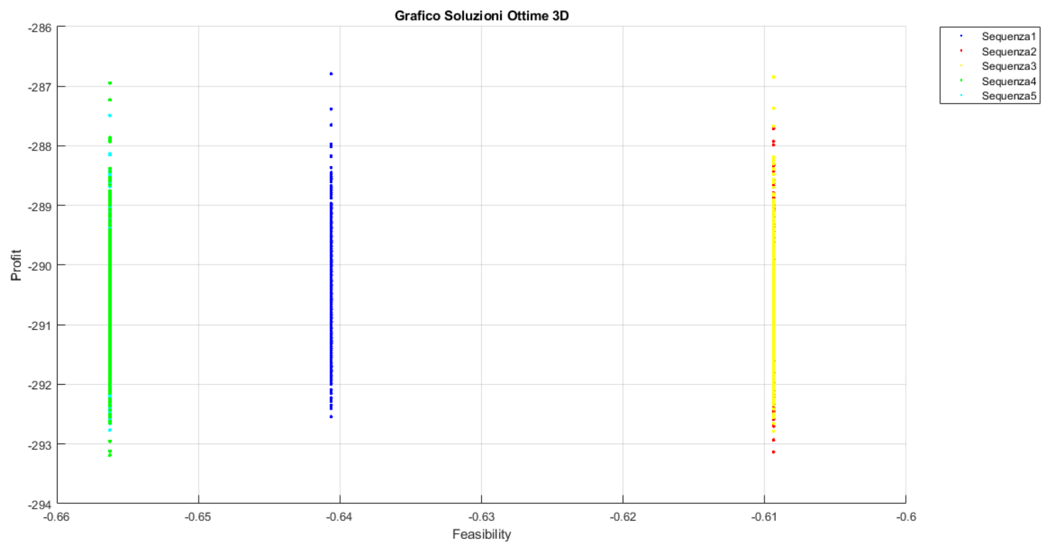


Figura 13: Grafico soluzioni ottime.

Nella Figura 12 possiamo vedere vari punti, i quali rappresentano le varie soluzioni con i tempi perturbati. Nella Figura 13 possiamo notare bene, come la feasibility non varia al variare del tempo, questo comporta che le soluzioni ottime, si dispongano lungo una perpendicolare verso tale asse.

5.6.1 Valutazione della robustezza

Per capire quali tra le soluzioni ottime selezionate sia la più robusta, cioè soggetta a meno variazioni rispetto al suo valore ottenuto utilizzando il vettore dei tempi statici, ho utilizzato una rivisitazione dell' indice di Xie-Beni.

5.6.2 Indice Xie-Beni

L'indice di Xie-Beni, è un criterio di validità per misurare la compattezza del fuzzy cluster e la separazione tra i cluster. La compattezza è definita come

$$J_m = 1/N \sum_{i=0}^{c} \sum_{j=0}^{N} \mu_{ij}^{m} \, d^2(X_j, c_i)$$

dove μ_{ij}^{m} è il valore di appartenenza al cluster di un punto j, m è l'indice di fuzzy, $d^2(X_j, c_i)$ è la distanza tra il centro del cluster e il punto. La separazione tra cluster è indicata come la minima distanza tra i centri:

$$d_{min} = min_{ij}[d^2(c_i, c_j)]$$

Il rapporto tra compattezza e separazione è definito come indice di Xie-Beni (XB)

$$XB = J_m/d_{min}$$

La rivisitazione della formula, consiste nel vedere la minima distanza tra cluster uguale a 1, dato che una soluzione appartiene ad un solo cluster, in modo tale da vedere $XB = J_m$, che dipenda solo dalla compattezza,$mu_i j = 1$ per tutti i punti. La formula rivisitata è la seguente

$$J_m = 1/N \sum_{j=0}^{N} d^2(X_j, c_i) * \delta_j$$

quindi la compattezza è ottenuta dalla distanza media tra il centro del cluster, che in questo caso è il valore della soluzione con i tempi medi, e le varie sequenze ottenute perturbando i vari tempi, ogni distanza moltiplicata per δ_j, il quale rappresenta la probabilità che si verifichi con determinati tempi selezionati tale sequenza . Il codice di seguito, riporta come ciò sia stato implementato

```


vettor = zeros(66,7);

for j = 1:length(media)
vettor(media(j),:) = [2/27 3/27 14/27 3/27 2/27 2/27 1/27];

end

for j = 1:length(bassa)
vettor(bassa(j),:) = [1/27 1/27 13/27 4/27 4/27 3/27 1/27];

end

for j = 1:length(elevata)
vettor(elevata(j),:) = [1/27 2/27 14/27 4/27 3/27 2/27 1/27];

end
```

I tre vettori rappresentano, i vettori che vengono selezionati in base alla classe di difficoltà di appartenenza dell' oggetto da rimuovere, nel caso della rimozione della vite, veniva selezionato il vettore a bassa difficoltà.

6 Test

Ora si effettua un test cercando di trovare da un insieme di soluzioni ottime la più robusta. Quindi si inzia, lanciando il problema su PlatEMO e andando a creare una popolazione di soluzioni ottime, tra queste ne vengono scelte 5 aventi maggior profitto. Queste soluzioni vengono perturbate .

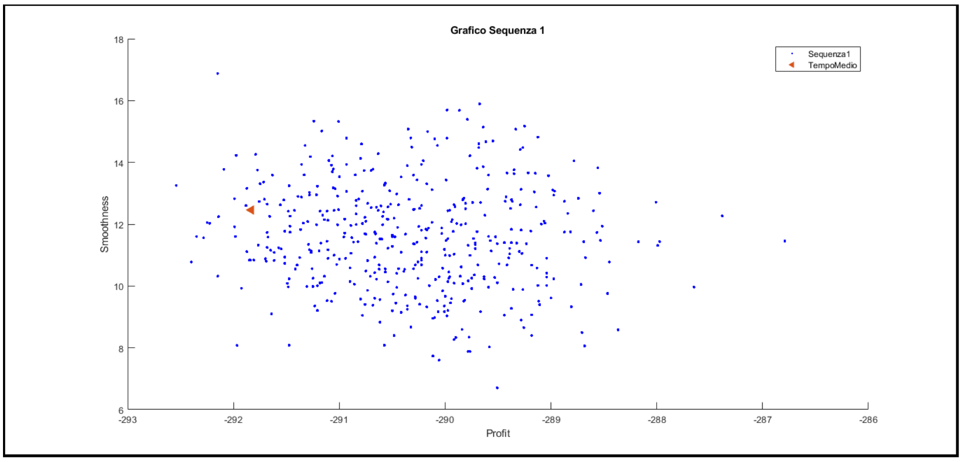


Grafico Sequenza 1
Sequenza1
TempoMedio
Smoothness
Profit
18
16
14
12
10
8
6
-293
-292
-291
-290
-289
-288
-287
-286

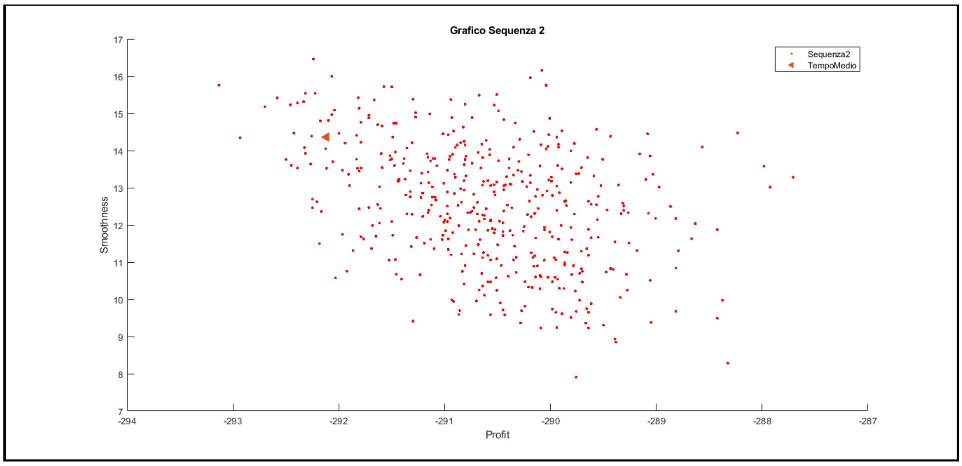


Grafico Sequenza 2
Sequenza2
TempoMedio
Smoothness
Profit
17
16
15
14
13
12
11
10
9
8
7
-294
-293
-292
-291
-290
-289
-288
-287

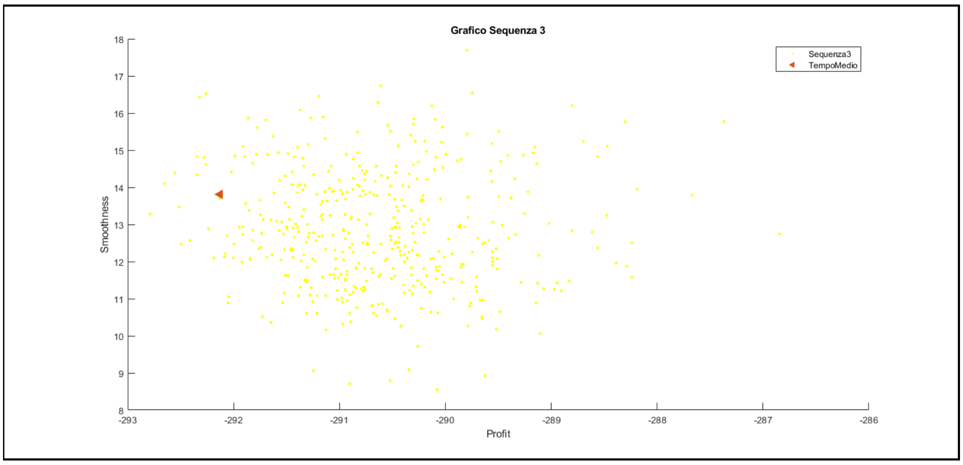
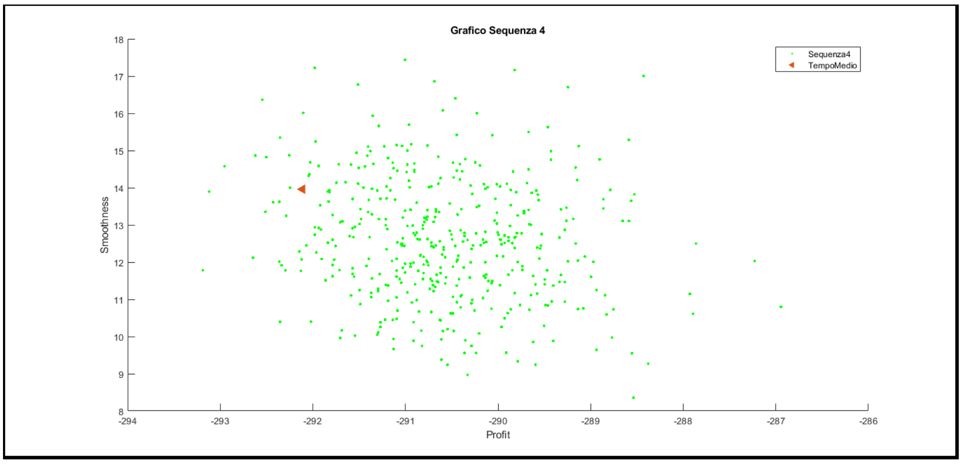


Grafico Sequenza 3
Sequenza3
TempoMedio
Smoothness
Profit
18
17
16
15
14
13
12
11
10
9
8
-293
-292
-291
-290
-289
-288
-287
-286



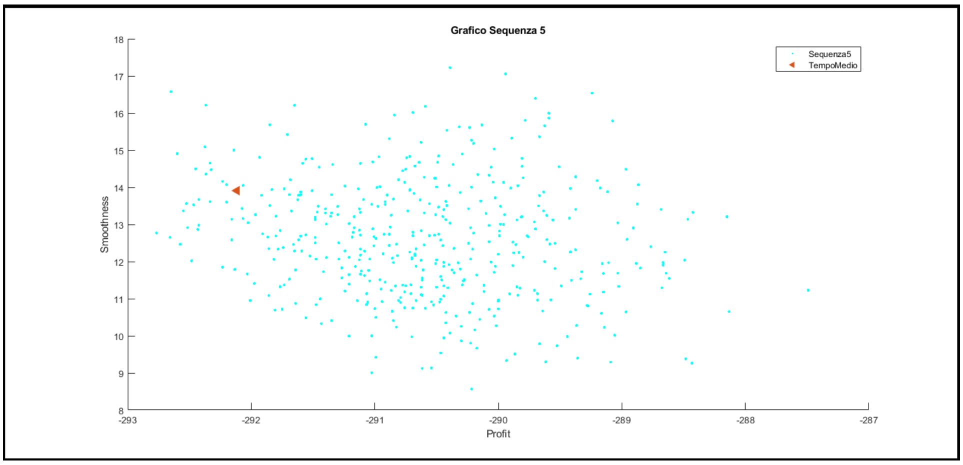


Figura 14: Rappresentazione delle 5 soluzioni ottime con i tempi perturbati

Succcessivamente viene calcolata la distanza tra la soluzione ottenuta utilizzando il vettore dei tempi medi e ogni soluzione ottenuta perturbando tali tempi. Sotto è riportato il codice implementato .

```
dist_CX = zeros(5,448);


 for i = 1:length(z1)
   dist_CX(1,i) = distance(mediop(1),medios(1),y1(i),z1(i));
 end
 for i = 1:length(z2)
   dist_CX(2,i) = distance(mediop(2),medios(2),y2(i),z2(i));
 end
 for i = 1:length(z3)
   dist_CX(3,i) = distance(mediop(3),medios(3),y3(i),z3(i));
 end
 for i = 1:length(z4)
   dist_CX(4,i) = distance(mediop(4),medios(4),y4(i),z4(i));
 end
 for i = 1:length(z5)
   dist_CX(5,i) = distance(mediop(5),medios(5),y5(i),z5(i));
 end
```

Infine si calcola l'indice Xie-Beni per ogni soluzione.

```

 index_XB = zeros(1,5);

 for j = 1:5
     for i = 1:448

         index_XB(j) = probvalue(5,j)*dist_CX(j,i);
     end



 end

 index_XB = index_XB/448;

 disp(index_XB);
 disp(min(index_XB));
```

Tra le soluzioni viene scelta come migliore, quella che presenta l'indice più piccolo, cioè la più robusta.

7 Conclusioni

Il metodo implementato riesce partendo da un pool di soluzioni ottime a trovare la migliore per robustezza, Questa scelta consente in una azienda adibita allo smontaggio di scegliere tra le soluzione più proficue quella che nel lungo termine rimane più costante con i tempi nell'esecuzione dell'operazioni di smontaggio.

8 Ringraziamenti

Siamo giunti alla fine, è stato un lungo cammino fatto di momenti di gioia, di sofferenza, di lunghe nottate passate sui libri,di tanti sacrifici , ma oggi a conclusione di questa mia esperienza,poso dire che n' è valsa la pena. Oltre che ad aver arricchito il mio bagaglio culturale, posso affermare di essere soprattutto cresciuto come persona, sono fiero di me stesso e di come ho affrontato questo cammino, certo non posso dire che a volte non ho fatto delle scelte sbagliate, ma è soprattutto grazie a queste che oggi sono qui. Ci tengo a ringraziare tutte le persone che mi hanno supportato. Parto quindi con il ringraziare i miei genitori , mi hanno sempre spinto nella direzione dello studio perchè erano consci delle mie capacità e del fatto che sarei riuscito ad arrivare dove sono ora.Spero di averli resi orgogliosi di me. Ringrazio il mio fratello Franz, è un importantissima parte di me, lo ringrazio soprattuto per il sostegno e il supporto che mi ha sempre dato. Ringrazio le mie sorelle, perchè sono state un esempio per me. Voglio ringraziare tutti i miei amici, nuovi e vecchi che ho conosciuto nella mia esperienza universitaria, abbiamo superato mille difficoltà insieme. In particolare ci tengo a ringraziare Leonardo, un amico che conoscevo già prima di intraprendere questa esperienza, il cui legame si è rafforzato ulteriolmente in questi anni. Lo ringrazio per tutte le volte in cui c'è stato, per tutte le volte in cui mi ha sostenuto e posso solo dire grazie. Ci tengo a ringraziare Davide, una persona che invece ho conosciuto in questa esperienza, lo ringrazio per tutto l'aiuto che mi ha dato,e per tutte le volte che con una battuta mi ha strappato un sorriso. Ringrazio Giulio un amico che ho conosciuto solamente negl' ultimi anni, ma è una delle persone più buone e pulite che io conosco. Ringrazio Mirko una persona che ammiro molto per la sua conoscenza dela mia materia, e lo ringrazio per la mano che mi ha dato. Voglio ringraziare tutti i professori che ho avuto, ognuno a suo modo ha arricchito il mio percorso,in particolare voglio ringraziare il mio relatore per la disponibilità che mi ha concesso e per le competenze che mi ha trasmesso. Ringrazio nuovamente tutti, non sarei qui senza di voi, spero che questo traguardo sia solo l'inizio di un bellissimo viaggio.