



**Politecnico  
di Torino**

## **Politecnico di Torino**

Corso di Laurea

A.a. 2021/2022

TESI DI LAUREA MAGISTRALE IN INGEGNERIA CIVILE

# **Ottimizzazione di strutture reticolari in acciaio attraverso l'utilizzo di EDA**

**Relatori:**

**Prof. Ing. Giuseppe Carlo Marano**

**Prof. Amedeo Manuello Bertetto**

**Correlatori:**

**Dott. Jonathan Melchiorre**

**Candidato:**

**Rosi Gabriele**



## ABSTRACT

Lo scopo di questa tesi di laurea magistrale è lo sviluppo e l'applicazione a problemi di natura strutturale di un algoritmo di ottimizzazione metaeuristico. Il metodo presentato rientra nel campo degli Evolution of Distribution Algorithm (EDA), in quanto l'ottimizzazione del problema è ottenuta mediante l'evoluzione di un modello probabilistico che descrive la probabilità di trovare l'ottimo in una data zona dello spazio di ricerca. In particolare, il modello probabilistico alla base del metodo è ottenuto attraverso il Gaussian Mixture Model (GMM), che permette di ottenere una funzione di densità di probabilità composta da modelli gaussiani opportunamente pesati sulla base della funzione obiettivo. Il metodo è pensato per essere applicato a problemi strutturali di natura vincolata e, per tale motivo, si è adottato un approccio basato sulla penalizzazione della funzione obiettivo. La "penalty function" si è, quindi, ottenuta attraverso una formulazione originale basata sulla distanza euclidea che intercorre tra le soluzioni "feasible" e le soluzioni "unfeasible" studiate. Infine, allo scopo di adattare il metodo proposto a problemi di diversa natura, l'algoritmo è stato implementato in maniera parametrica, lasciando la possibilità all'utente di modificare due parametri legati alla matrice di varianza-covarianza delle gaussiane che generano il GMM. In tale modo è possibile prediligere la fase iniziale di esplorazione del dominio o la fase finale di ricerca della soluzione di ottimo, a seconda della natura del problema studiato. Nella seconda parte della tesi, l'efficacia del metodo presentato è stata provata attraverso le applicazioni su problemi benchmark di tipo analitico e strutturale e la comparazione con altri algoritmi di ottimizzazione noti. Il lavoro svolto ha permesso lo sviluppo di un tool che permette di utilizzare l'algoritmo all'interno del software Grasshopper, un programma che permette il "visual coding" nell'ambiente di progettazione 3D chiamato Rhinoceros. Tale tool è, infine, stato utilizzato in collaborazione con lo studio di progettazione Maffei Engineering SpA per l'ottimizzazione di una struttura reticolare in acciaio reale.

# INDICE

<b>1</b>	<b>INTRODUZIONE.....</b>	<b>6</b>
	1.1 GLI ALGORITMI DI OTTIMIZZAZIONE METAEURISTICI .....	8
	1.2 PERCHÉ GLI EDA?.....	9
	1.3 STRUTTURA DELLA TESI.....	10
<b>2</b>	<b>EDA .....</b>	<b>11</b>
	2.1 PROCEDURA.....	12
	2.2 CAMPIONAMENTO INIZIALE .....	14
	2.3 SELEZIONE DEGLI INDIVIDUI.....	18
	2.4 MODELLO PROBABILISTICO .....	20
	2.4.1 Peso .....	24
	2.4.2 Varianza .....	25
	2.4.3 Codice .....	27
<b>3</b>	<b>PROBLEMI NON VINCOLATI.....</b>	<b>29</b>
	3.1 MODALITÀ.....	30
	3.2 SEPARABILITÀ .....	30
	3.3 DEFINIZIONE DEL PROBLEMA .....	31
	3.4 TEST DI BENCHMARK .....	32
	3.4.1 Funzione Sferica.....	33
	3.4.2 Funzione di Himmelblau .....	36
	3.4.3 Funzione di Ackley.....	38
	3.4.4 Funzione di Rastrigin.....	41
<b>4</b>	<b>PROBLEMI VINCOLATI.....</b>	<b>44</b>
	4.1 DEFINIZIONE DEL PROBLEMA .....	44

4.2	DEFINIZIONE DELLA PENALITÀ .....	45
4.3	TEST DI BENCHMARK .....	48
4.3.1	Problema di Sickle .....	48
4.3.2	Truss 10 .....	51
5	CASO STUDIO: STRUTTURA RETICOLARE IN ACCIAIO .....	56
5.1	INTRODUZIONE .....	56
5.2	DESCRIZIONE CASO STUDIO .....	58
5.2.1	Schema statico.....	60
5.2.2	Materiale .....	60
5.2.3	Carichi .....	61
5.2.4	Combinazioni.....	61
5.2.5	Verifiche .....	62
5.3	MODELLO PARAMETRICO .....	63
5.3.1	processo di ottimizzazione.....	64
5.3.2	Parametrizzazione geometrica.....	67
5.3.3	Parametrizzazione modello FEM.....	69
5.3.4	Ottimizzatore .....	73
5.4	CASI STUDIO.....	78
5.4.1	Caso studio 1.....	79
5.4.2	Caso studio 2.....	96
5.4.3	Risultati e confronto .....	111
6	CONCLUSIONI.....	116
6.1	SVILUPPI FUTURI .....	117
7	BIBLIOGRAFIA .....	118

# 1 INTRODUZIONE

Lo scopo primario di ogni progettista è la ricerca di soluzioni economiche e affidabili, rispettando i vincoli architettonici, normativi e ambientali. L'esigenza di strumenti che aiutino a sopperire a tali richieste ha reso ormai indispensabile la conoscenza di software sofisticati e modelli algoritmici complessi. L'impatto di tali strumenti sulla progettazione ha permesso di analizzare una quantità di soluzioni impossibile da vagliare per un solo uomo, soprattutto in problemi strutturali complessi e di grosse dimensioni. Ciò ha reso necessario la nascita di una metodologia che possa fornire al progettista un'arma in più nell'affrontare questa eterna sfida della ricerca dell'ottimo strutturale. Per tali scopi sono stati creati algoritmi di ottimizzazione che ispirano il loro funzionamento a processi di ricerca dell'ottimo presenti in natura. Un esempio sono gli algoritmi genetici che attraverso un meccanismo di selezione e rimescolamento dei caratteri degli individui, permettono la sopravvivenza del solo individuo più prestante. Ma utilizzare questi strumenti a scatola chiusa su problemi strutturali limita lo sfruttamento a pieno delle loro potenzialità poiché, ogni problema è unico e non esiste un unico solutore per tutti i problemi. Si è reso indispensabile quindi conoscere a pieno questi strumenti, capire a fondo il loro modo di agire nei confronti delle strutture e impostarli in funzione del problema che si affronta.

La seguente tesi di laurea magistrale è il risultato della collaborazione tra il Dipartimento di Ingegneria Strutturale, Edile e Geotecnica del Politecnico di Torino e lo studio di progettazione strutturale Maffei Engineering SpA.

Lo scopo della tesi è lo sviluppo e l'applicazione, su strutture reticolari, di un algoritmo evolutivo probabilistico EDA, basato sul modello di mistura delle gaussiane detto anche Gaussian Mixture Model.

Il percorso attraversato per raggiungere tale scopo ha previsto una fase iniziale di implementazione e test delle varie funzionalità dell'algoritmo su test di benchmark a difficoltà crescente. In primis sono stati affrontati test su problemi non vincolati al fine di implementare le impostazioni base dell'EDA quali la matrice di varianza e la funzione di redistribuzione dei pesi. Successivamente sono stati condotti una serie di test su funzioni vincolate grazie alle quali è stata testata una innovativa funzione di penalizzazione basata sulla distanza euclidea tra gli individui generati dall'algoritmo. Il codice di programmazione dell'EDA è stato implementato sia in linguaggio Matlab che in linguaggio Python.

Una volta completato e testato l'EDA e le sue funzionalità è stato possibile applicare l'algoritmo proposto ad un caso studio reale, in particolare a una trave reticolare in acciaio di luce 300 metri. Per raggiungere tale scopo è stato creato un modello parametrico all'interno del software Grasshopper, un programma che permette il "visual coding" nell'ambiente di progettazione 3D chiamato Rhinoceros. All'interno di Grasshopper utilizzando la tool FeMM sviluppata internamente da Maffei Engineering SpA è stato creato un modello unico parametrico legato al programma agli elementi finiti Sap2000. Inoltre, sempre all'interno di Grasshopper, è stata sviluppata una tool che permettesse l'utilizzo dell'algoritmo EDA internamente al programma, permettendo così di condurre un processo di ottimizzazione sia geometrico che strutturale.

## 1.1 GLI ALGORITMI DI OTTIMIZZAZIONE METAEURISTICI

“Secondo Glover e Kochenberger (2006), la metaeuristica può essere definita come "un processo iterativo che guida il funzionamento di una o più subordinate euristiche (che possono andare da un processo di ricerca locale a un processo costruttivo di soluzioni casuali) per produrre in modo efficiente soluzioni di qualità per un problema". Le metaeuristiche sono una famiglia di tecniche di ottimizzazione non lineare approssimata che forniscono soluzioni accettabili (in genere quasi ottimali), in un tempo ragionevole, per risolvere problemi di computazione difficili e complessi in campo scientifico, ingegneristico e in altri campi. A differenza degli algoritmi di algoritmi di ottimizzazione esatti, le metaeuristiche non garantiscono soluzioni ottimali dimostrabili.

Tuttavia, per molti problemi reali su larga scala, la metaeuristica può essere preferita ai metodi basati sul gradiente o alla programmazione matematica (Singh e Jana, 2017). Lo stesso vale nel caso di problemi di ottimizzazione con funzioni obiettivo non lisce (Juan et al., 2020). Esistono anche metodi efficaci basati sul gradiente, come quello dell'approssimazione stocastica a perturbazione simultanea (Spall, 2005). Questi metodi sono adatti alla modellazione e all'ottimizzazione adattiva in condizioni di incertezza (Bhatnagar et al., 2003) e all'ottimizzazione del controllo (Li, Jafarpour e Mohammad-Khaninezhad, 2013). Tuttavia, questi metodi mostrano limitazioni in presenza di funzioni obiettivo non lisce (come quelle dovute all'esistenza di vincoli morbidi realistici), dove i gradienti non possono essere facilmente calcolati. Le metaeuristiche, invece, sono metodi di ottimizzazione privi di derivati.

Le metaeuristiche possono essere classificate in base a varie caratteristiche (Talbi, 2009): ispirate alla natura o meno, deterministiche o stocastiche, basate sulla popolazione o sulla singola soluzione, iterative o avide, ecc. Un altro aspetto da tenere in considerazione nella scelta di un meta-euristico è la sua capacità di esplorazione rispetto a quella di sfruttamento. Questo concetto è solitamente legato a diverse sottofamiglie. Così, mentre gli algoritmi basati su una singola soluzione manipolano e trasformano una singola soluzione durante la ricerca (alta intensificazione), gli algoritmi basati sulla popolazione evolvono un'intera popolazione di soluzioni (alta diversificazione). Le metaeuristiche basate su una singola soluzione possono essere viste come "passeggiate" attraverso quartieri o traiettorie di ricerca nello spazio di ricerca del problema in questione. Vengono eseguite mediante procedure iterative che si spostano dalla soluzione corrente a un'altra basata su metodi di ricerca locale. Tra le metaeuristiche più importanti di questa sottofamiglia vi sono: la ricerca tabu (Glover e Laguna, 2013), l'annealing

simulato (Kirkpatrick, Gelatt e Vecchi, 1983), la ricerca per prossimità variabile (Hansen, Mladenovic e Moreno, 2010), la procedura di ricerca adattiva randomizzata greedy, o GRASP (Feo e Resende, 1995), e la ricerca locale iterata (Lourenc,o, Martin e Stutzle, 2010). Nell'insieme delle metaeuristiche basate sulla popolazione, gli algoritmi evolutivi e, in particolare, gli algoritmi genetici sono frequentemente utilizzati in molti problemi di ingegneria e produzione (Lee, 2018). Esistono molti altri algoritmi che si basano sulla gestione di un insieme di soluzioni a ogni iterazione. Si tratta, tra gli altri, dell'ottimizzazione a formica (Dorigo e Stutzle, 2004), dell'ottimizzazione a sciame di particelle (Kennedy, 2010), della ricerca per dispersione (Laguna e Marti, 2012) e degli algoritmi di stima della distribuzione (Larranaga e Lozano, 2002). Infine, gli algoritmi memetici (Moscato e Mathieson, 2019) possono essere visti come un connubio tra la metaeuristica basata sulla popolazione e la metaeuristica a soluzione singola.” [1]

## **1.2 PERCHÉ GLI EDA?**

Tra l'ampio panorama degli algoritmi metaeuristici sono stati scelti gli Estimation of distribution algorithms EDA basati sulla popolazione, per le loro potenzialità applicative su problemi di natura strutturale. In particolare, l'approccio dell'EDA, alla soluzione del problema di ricerca dell'ottimo, lo porta a non scartare mai le informazioni derivate da ogni singolo individuo analizzato dall'algoritmo, anche per un individuo lontano dalla frontiera dei vincoli imposti dal problema specifico. Ciò è di rilevante utilità nel campo della progettazione strutturale dal momento che l'onere computazionale derivato dalle analisi strutturali è molto alto. Per tale motivo è nato il bisogno di utilizzare un algoritmo che possa arrivare alla soluzione di ottimo analizzando quanti meno individui possibile. Per fare ciò quindi ogni informazione derivata dalla popolazione analizzata diventa fondamentale, soprattutto per descrivere la frontiera della zona “feasible”. Inoltre, questa particolare peculiarità rende gli EDA adatti ai problemi strutturali vista la vicinanza dell'ottimo globale di una struttura alla zona di confine tra individui “feasible” e “unfeasible”.

### **1.3 STRUTTURA DELLA TESI**

La tesi è composta da 5 capitoli, senza considerare il capitolo attuale.

Nel capitolo 2 è introdotto il funzionamento del EDA e la sua implementazione in codice Matlab e Python.

Nel capitolo 3 sono introdotti e risolti i test di benchmark per funzioni non vincolate con un confronto dei risultati con altri tipi di algoritmi metaeuristici.

Nel capitolo 4 sono introdotti e risolti i test di benchmark per funzioni vincolate con un confronto dei risultati con altri tipi di algoritmi metaeuristici. Inoltre, è presentata la teorizzazione per una nuova funzione di penalizzazione.

Nel capitolo 5 è affrontato problema di tipo strutturale in collaborazione con Maffei Engineering SpA riguardante il problema di ottimizzazione di una trave reticolare in acciaio utilizzando l'EDA.

Nel capitolo 6 sono esposte le conclusioni del lavoro svolto.

## 2 EDA

Gli Estimation of distribution algorithms (EDAs) sono algoritmi evolutivi probabilistici nel quale l'ottimizzazione è vista come una serie di aggiornamenti incrementali di un modello probabilistico, a partire dal modello che codifica la distribuzione uniforme delle soluzioni ammissibili, e terminando con il modello che genera solo l'optima globale.

Gli EDAs, oltre a fornire un ottimo strumento per risolvere problemi di ottimizzazione, affrontando la ricerca dell'ottimo con un modello computazionale compatto, hanno il vantaggio di fornire molte informazioni sul dominio del problema interessato. Le informazioni rilasciate dal modello probabilistico permettono di creare strumenti per velocizzare e influenzare la ricerca dell'ottimo in vari aspetti [2].

Un problema di ottimizzazione generico può essere definito descrivendo un insieme delle potenziali soluzioni al problema e attuando una procedura che determini la qualità di queste soluzioni. L'insieme delle potenziali soluzioni è rappresentato da un sistema generale di valutazione delle soluzioni ammissibili e da una serie di vincoli che vadano a escludere o penalizzare le soluzioni non ammissibili. La procedura per la valutazione della qualità delle soluzioni candidate può essere definita come una funzione "obiettivo", la quale debba essere minimizzata o massimizzata a seconda del problema. Quindi la finalità dell'algoritmo sarà quella di trovare una soluzione dall'insieme degli individui ammissibili che massimizzi la qualità di essi, come definito dalla procedura di valutazione.

Attraverso i tre passaggi seguenti viene riassunto il metodo generico per la ricerca dell'ottimo di un problema generico di ottimizzazione:

- Generare soluzioni candidate.
- Valutare le soluzioni generate.
- Aggiornare la procedura per la generazione dei nuovi candidati in funzione dei risultati della valutazione precedente.

La qualità delle soluzioni generate tende nel tempo a migliorare trovando, dopo un adeguato numero di iterazioni, l'ottimo globale o la sua approssimazione accurata [2].

## 2.1 PROCEDURA

Il concetto di fondo degli EDA è costruire e aggiornare un modello probabilistico che rappresenti una distribuzione di probabilità rispetto alle soluzioni candidate. Ad ogni iterazione, il modello muta in funzione dei risultati della valutazione dei candidati generati in modo tale da generare una soluzione migliore nelle successive iterazioni.

La differenza di fondo degli EDA dalle altre metodologie metaeuristiche, come gli algoritmi genetici, risiede nell'utilizzo di un modello probabilistico esplicito. Infatti, negli algoritmi genetici classici la distribuzione di probabilità utilizzata per generare le nuove soluzioni ammissibili sono spesso definite implicitamente da un operatore di ricerca o da una combinazione di più operatori [3].

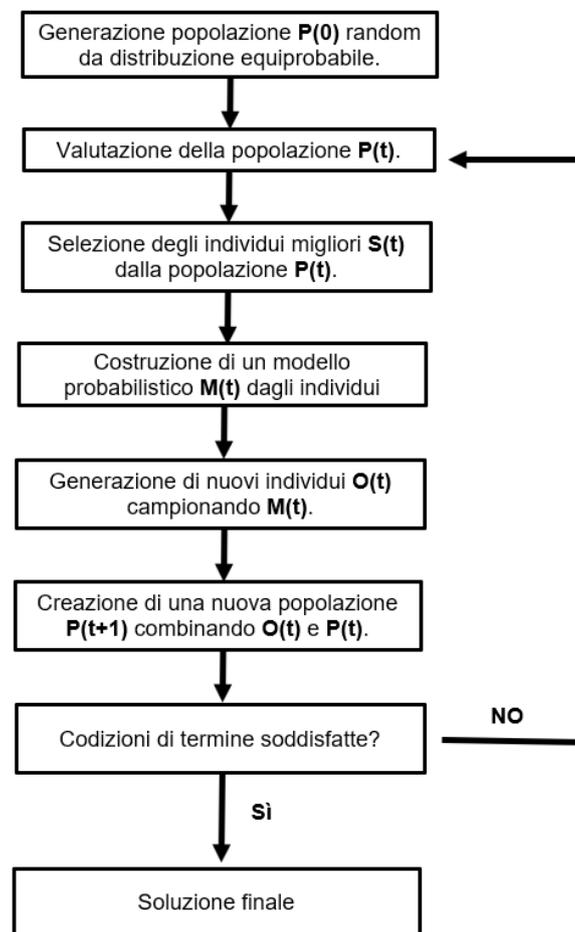


Figura 1 Diagramma di flusso dell'algoritmo EDA.

L'algoritmo proposto nella seguente tesi appartiene alla famiglia dei Population-Based EDAs che si distinguono nell'aver una prima fase nella quale si genera una popolazione di candidati random secondo una distribuzione probabilistica uniforme su tutto il dominio interessato. Ogni iterazione inizia creando una nuova popolazione di soluzioni utilizzando l'operatore di selezione

a troncamento, che dà la preferenza alle soluzioni di miglior qualità. Dopodiché dagli individui selezionati viene aggiornato il modello probabilistico per la generazione di una nuova popolazione e iterato il processo fino al raggiungimento dell'ottimo [3].

Per riassumere, i componenti principali di un Population-Based EDAs sono:

1. Un operatore di selezione per la scelta delle soluzioni più promettenti.
2. Una classe di modelli probabilistici da utilizzare per il campionamento della popolazione.
3. Una procedura per l'apprendimento di un modello probabilistico dalle soluzioni selezionate.
4. Una procedura per il campionamento di una nuova popolazione dal modello probabilistico costruito.
5. Un operatore sostitutivo per combinare le vecchie popolazioni e le nuove soluzioni candidate.

## 2.2 CAMPIONAMENTO INIZIALE

Lo step iniziale prevede il campionamento random di individui con una densità di probabilità equi distribuita su tutto il dominio del problema affrontato. Per fare ciò è stato utilizzato un generatore di numeri pseudo-casuali Mersenne Twister richiamabile nelle librerie di Python e Matlab dalla libreria “random”. L’utilizzo di tale algoritmo però portava a creare una mole di individui iniziale molto alta, ciò dovuto a una scarsa distribuzione degli individui sul dominio del problema.

Lo strumento utilizzato per risolvere tale problema è il Latin Hypercube Sampling (LHS); esso è un metodo statistico per generare un campione quasi casuale di valori dei parametri da una distribuzione multidimensionale.

Per definizione una griglia quadrata, contenete le posizioni dei campioni, è definita come Latin Hypercube se, e solo se, su ogni riga e colonna della griglia è presente un solo campione; trasponendo questo concetto a N dimensioni possiamo così definire un ipercubo.

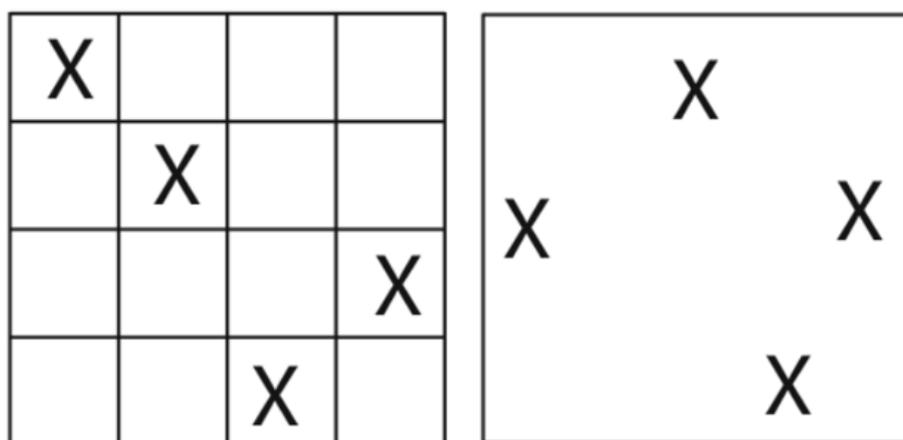


Figura 2 A sinistra campionamento dominio con Latin Hypercube, a destra campionamento random.

La procedura prevede che nel campionamento una funzione di N variabili, il dominio di ciascuna variabile sia diviso in M intervalli equi probabili. A questo punto M punti di campionamento vengono quindi posizionati per soddisfare i requisiti del Latin Hypercube; costringendo il numero di divisioni M, ad essere uguale per ogni variabile [4].

Possiamo quindi affermare che il Latin Hypercube assicura che l'insieme di numeri casuali campionati sia rappresentativo della variabilità reale del dominio mentre il campionamento casuale tradizionale è solo un insieme di numeri casuali senza alcuna garanzia.

Negli algoritmi population-based, la distribuzione degli individui della popolazione iniziale è strettamente collegata con il risultato finale. Questo aspetto è particolarmente evidente nel caso

del problema di ottimizzazione non vincolato della funzione di Himmelblau nella quale la funzione obiettivo racchiude la difficoltà di descrivere i quattro minimi assoluti ad essa appartenenti. Il problema è così definito:

$$\text{minimize } f(x) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \quad -6 \leq x, y \leq 6$$

$$\text{Min} = \begin{cases} f(3.0, 2.0) & = 0.0 \\ f(-2.805118, 3.131312) & = 0.0 \\ f(-3.779310, -3.283186) & = 0.0 \\ f(3.584428, -1.848126) & = 0.0 \end{cases}$$

Dove  $f(x)$  è la funzione obiettivo sulla quale ricercare il minimo della funzione,  $x$  e  $y$  sono le variabili del problema.

Sono state lanciate 50 analisi indipendenti con una popolazione di partenza pari a 100 individui e è stata calcolata la deviazione standard della popolazione dopo 100 iterazioni.

	<b>LHS</b>	<b>Random selection</b>
std.dev.	1.141521	1.068466

I risultati dimostrano come il LHS permetta di creare un campione di partenza più efficace riuscendo a identificare con maggior precisione i minimi locali.

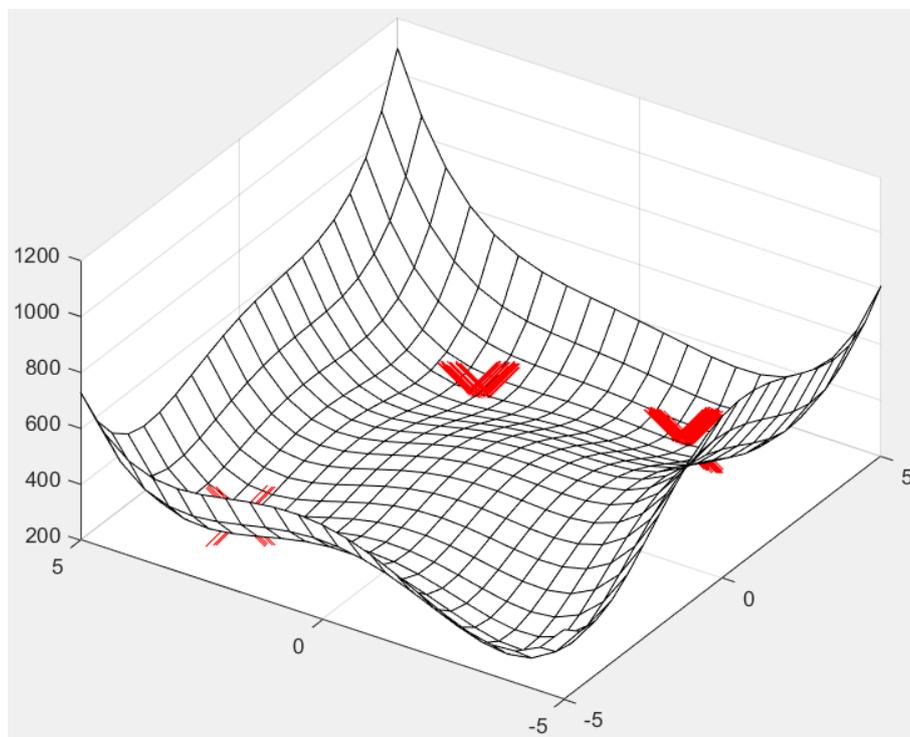


Figura 3 Generatore di numeri random: popolazione di partenza 100 individui.

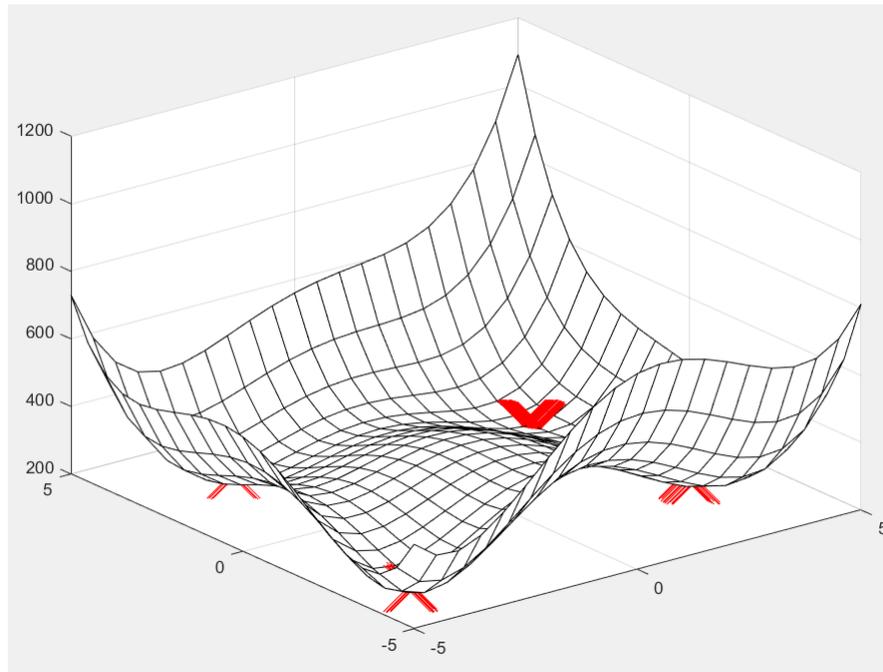


Figura 4 Generatore di numeri random: popolazione di partenza 200 individui.

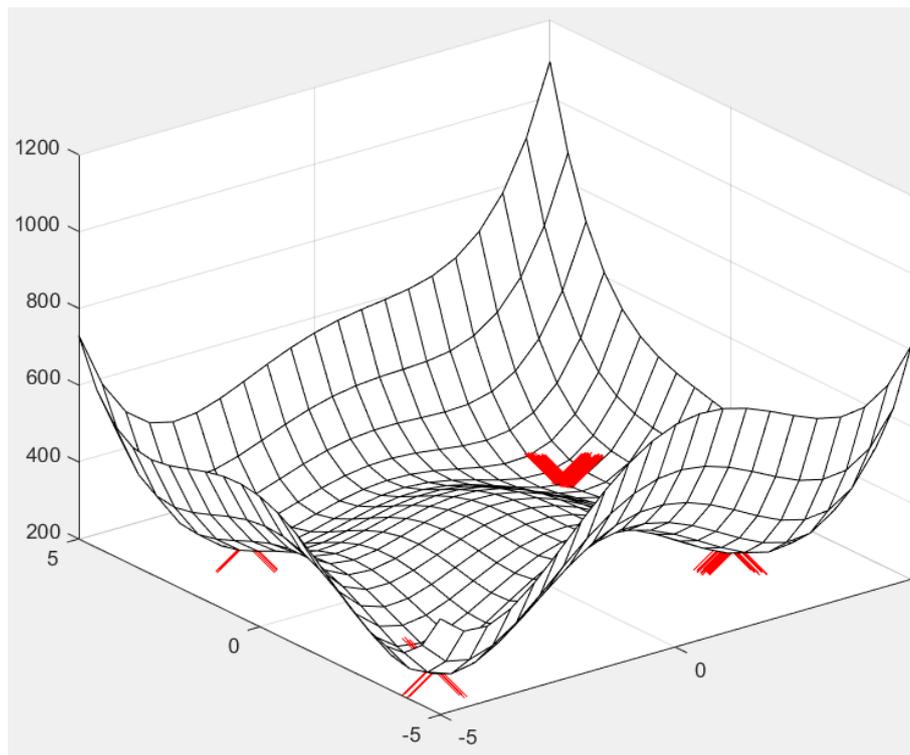


Figura 5 Latin hypercube sampling: popolazione di partenza 100 individui.

È stato possibile implementare questo strumento in Matlab attraverso il comando “*lhsdesign(n,p)*”, dove “n” e “p” sono le dimensioni del campione, e in linguaggio Python grazie alla libreria Scipy e alle seguenti righe di codice:

```
from scipy.stats import qmc
```

```
sampler = qmc.LatinHypercube(d)
```

```
sample = sampler.random(n)
```

dove “d” sono le dimensioni del campione e “n” il numero di individui da campionare.

### 2.3 SELEZIONE DEGLI INDIVIDUI

A seguito della valutazione del campione di individui generato, si opera una selezione di questi grazie alla semplice disposizione crescente o decrescente delle valutazioni, a seconda che l'obiettivo dell'ottimizzazione sia massimizzare o minimizzare la funzione obiettivo. Disponendo gli individui in base alla loro valutazione possiamo quindi selezionare, attraverso il metodo del troncamento, una percentuale definita di essi, in genere il 50%, sulla quale si baserà la costruzione del modello probabilistico.

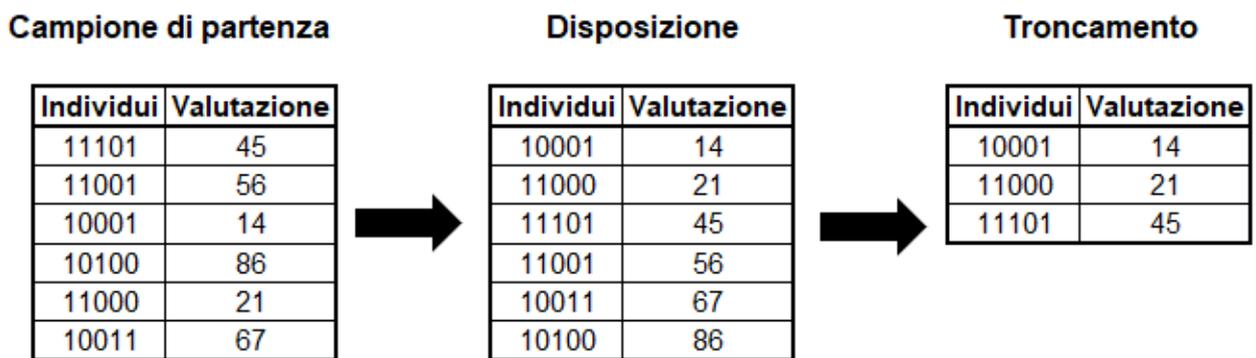


Figura 6 Selezione degli individui: disposizione e troncamento del campione

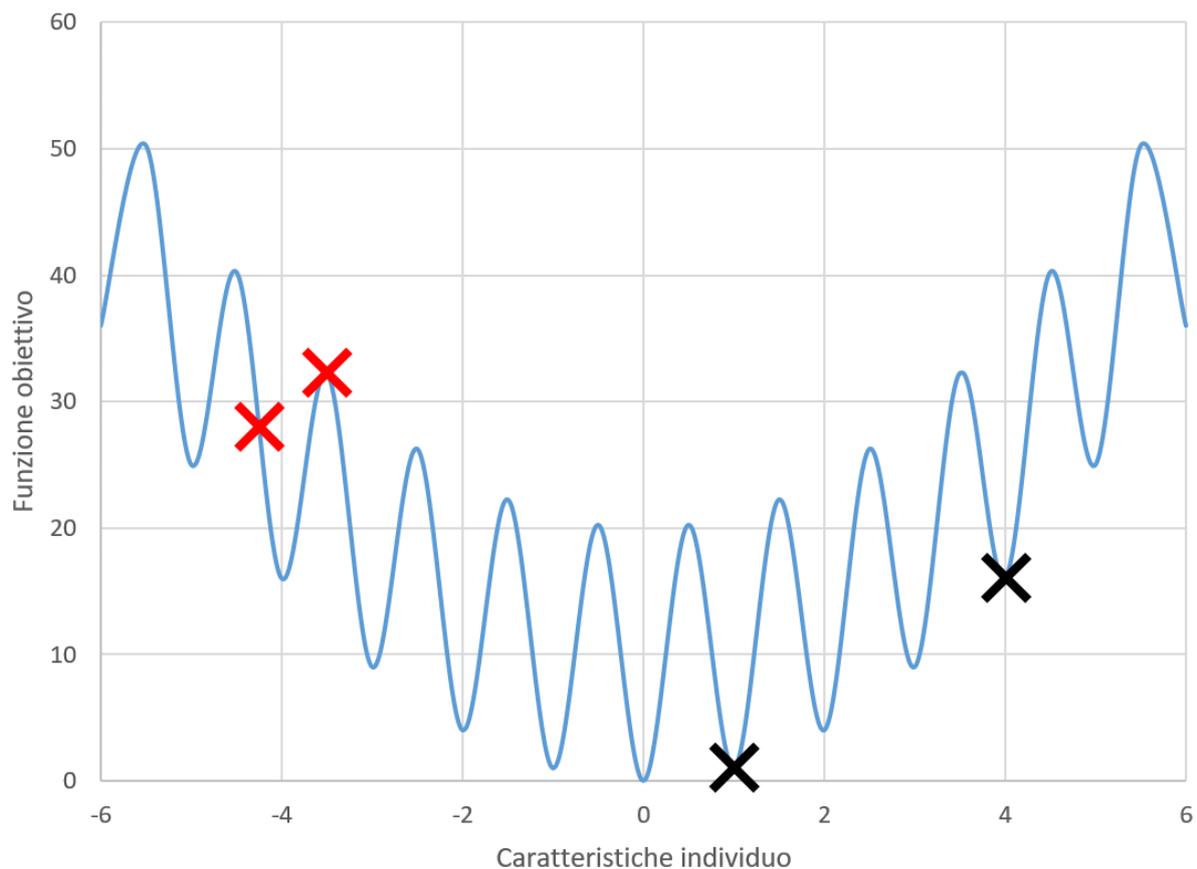


Figura 7 Esempio 1D su funzione di Rastring: in rosso gli individui scartati, in nero selezionati.

È stato possibile implementare il metodo in linguaggio Matlab grazie al comando `sortrows(A, column)` dove “A” si riferisce alla matrice da riordinare e “column” alla colonna delle valutazioni; per quanto riguarda il linguaggio Python è stato utilizzato il comando della libreria Numpy `argsort(A, asse)`, il quale restituisce un vettore di indici della stessa forma del vettore “A”, oggetto della disposizione, i quali indicizzano i dati lungo l’asse di riferimento.

## 2.4 MODELLO PROBABILISTICO

Il modello probabilistico proposto per la ricerca del punto di ottimo si basa su una distribuzione di probabilità generata da una miscela di distribuzioni normali chiamato “Gaussian Mixture Model” (GMM) [5]. Il “Gaussian Mixture Model” è uno strumento semi-parametrico in grado di descrivere distribuzioni sconosciute e complesse, là dove l’approccio parametrico fallisce.

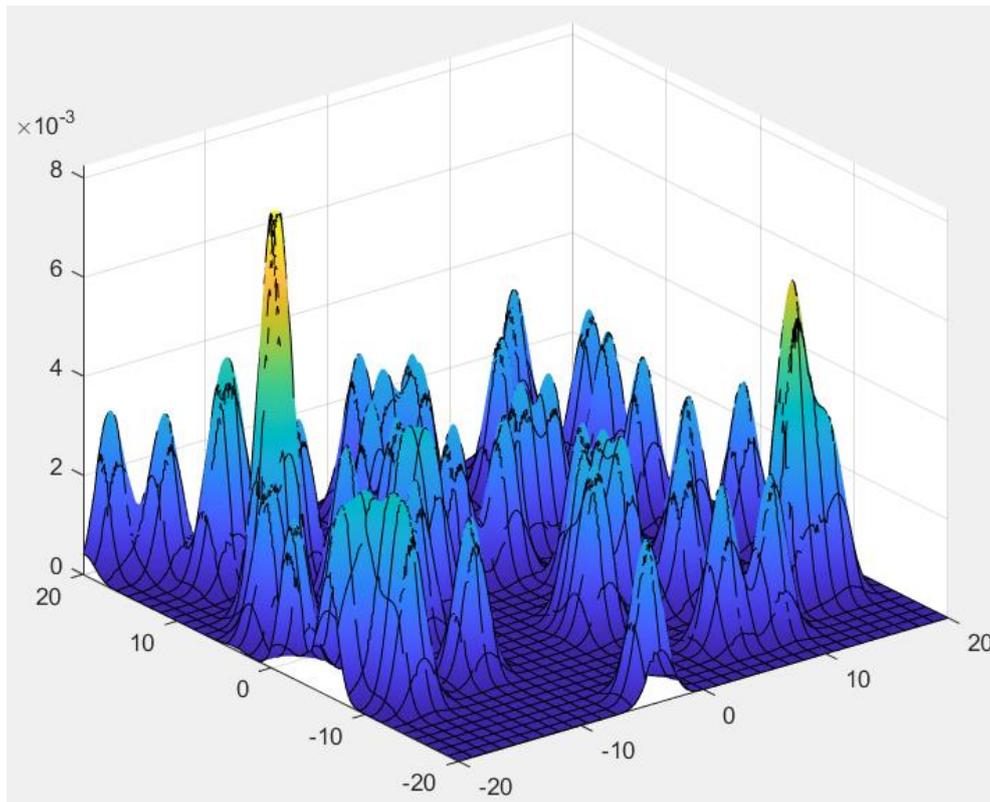


Figura 8 Modello GMM su dominio bidimensionale.

Il modello GMM è generato grazie alla sovrapposizione lineare di n densità di probabilità pari al numero dei degli individui selezionati:

$$N(x|\mu_i, \Sigma_i) = \frac{\exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i)\right)}{\sqrt{(2\pi)^d |\Sigma|}}$$

Dove:

- $d$ = dimensioni del candidato
- $\Sigma$  = matrice di varianza-covarianza
- $\mu_i$ = media

Ogni distribuzione normale è poi pesata in modo tale che la sommatoria dei pesi sia pari all’unità:

$$p(x) = \sum_{i=1}^n \pi_i \cdot p_i = \sum_{i=1}^n \pi_i \cdot N(x|\mu_i, \Sigma_i)$$

$$\text{con} \quad \sum_{i=1}^n \pi_i = 1, \quad 0 \leq \pi_i \leq 1$$

Dove:

- $p_i$  = probabilità individuo i-esimo
- $\pi_i$  = peso dell'individuo i-esimo

Impostando quindi la media  $\mu_i$  con le caratteristiche delle soluzioni  $x$  selezionate nel dominio di progetto, e una matrice di varianza-covarianza diagonale  $\Sigma$  di dimensioni  $d \times d$  per ogni elemento della popolazione, la GMM implementata consiste in una distribuzione normale multivariata con deviazioni standard arbitrarie per ogni variabile[nota2]. In questo caso, l'attuale GMM funziona in modo simile a una stima della densità Kernel (KDE) non parametrica [5].

Nelle seguenti immagini è mostrato con un esempio a una dimensione il processo di generazione del modello probabilistico passando quindi da una distribuzione costante della densità di probabilità a un modello GMM [6].

I parametri in entrata per la generazione del modello probabilistico sono quindi le caratteristiche  $\mu_i$  dell'individuo selezionato, il peso  $\pi_i$ , e la matrice di varianza-covarianza diagonale  $\Sigma$  di dimensioni  $d \times d$ .

L'obiettivo del modello Gaussiano è quello di studiare l'intorno dei candidati più promettenti, non tralasciando però l'esplorazione del dominio, in modo tale di non cadere in un falso minimo.

Nelle seguenti immagini si può vedere una rappresentazione della generazione della nuova distribuzione di probabilità considerando un problema mono-dimensionale.

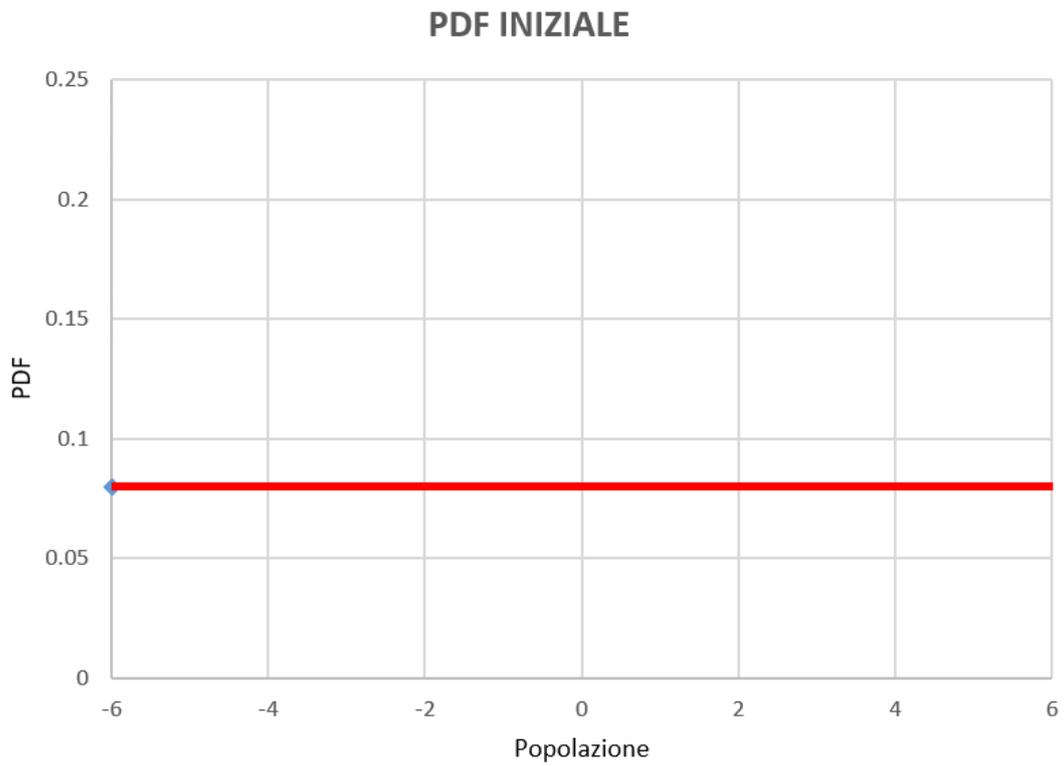


Figura 9 Distribuzione di probabilità iniziale costante.

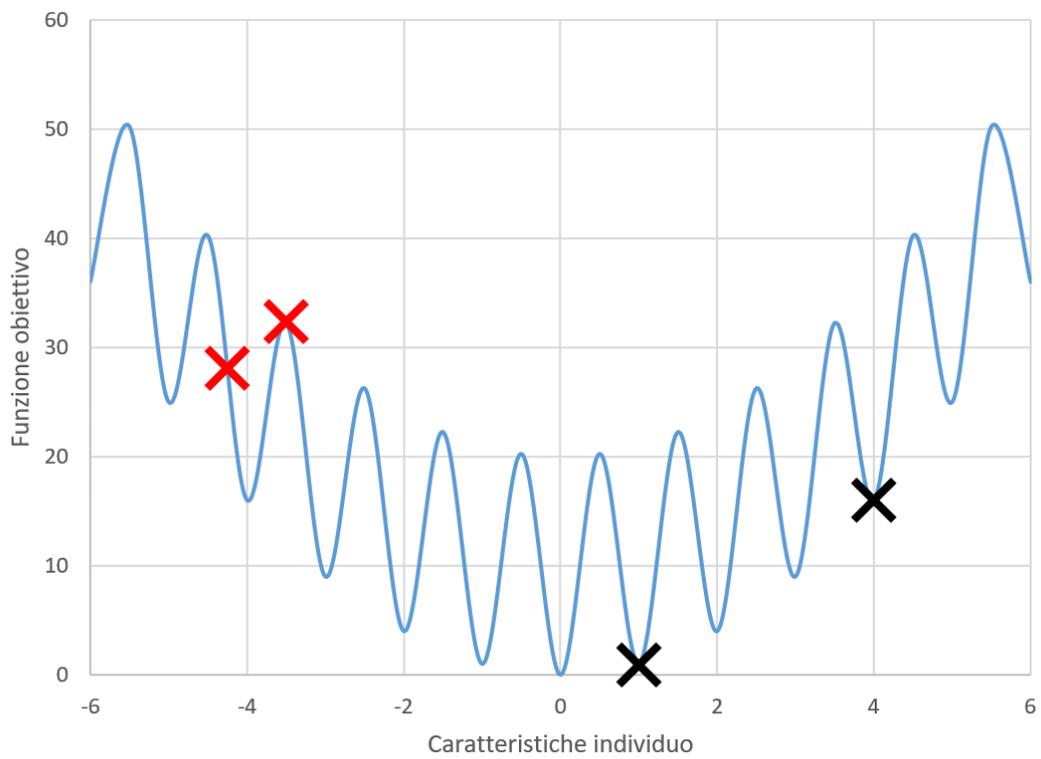
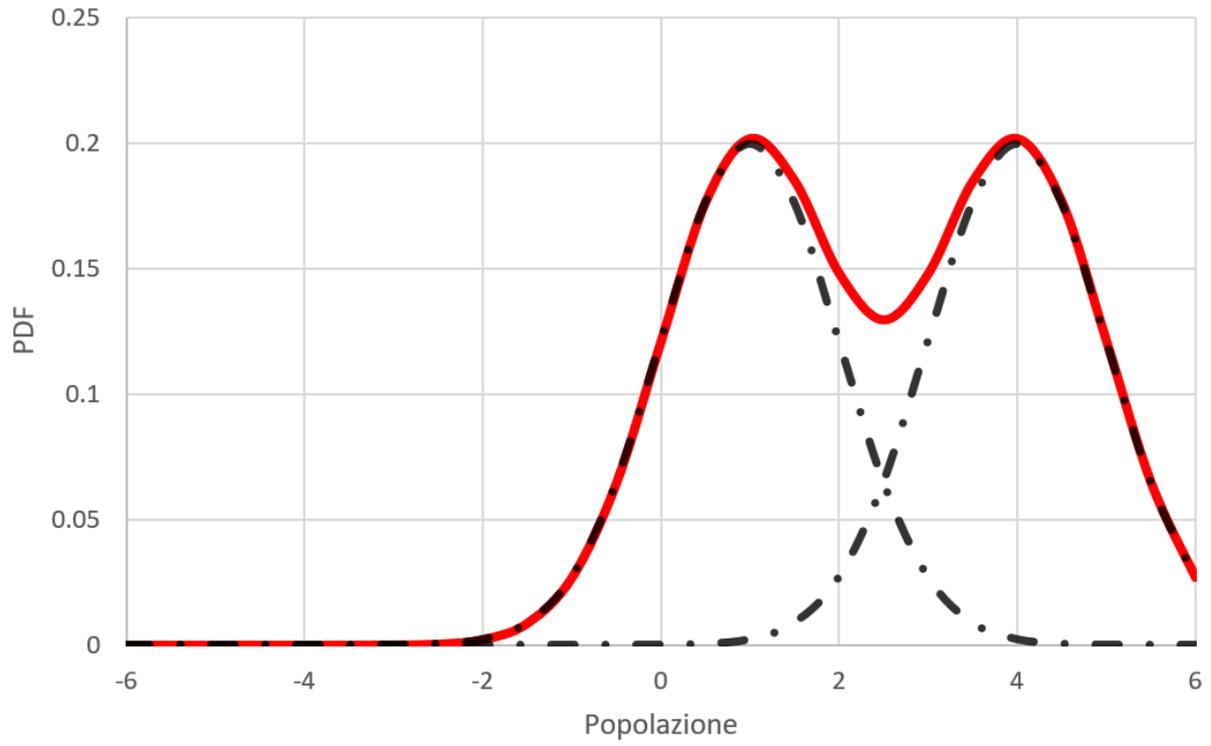


Figura 10 Generazione e selezione nuova popolazione: in rosso gli individui scartati, in nero gli individui selezionati.

## GMM PDF



*Figura 11 Gaussian Mixture Model: generazione del modello dagli individui selezionati*

### 2.4.1 PESO

La pesatura delle gaussiane  $\pi_i$  permette di affidare una maggiore densità di probabilità agli individui più promettenti, così che le nuove generazioni saranno estratte con caratteristiche più simili a quegli individui. Ciò si ripercuote sulla distribuzione con un incremento in altezza delle gaussiane con maggior peso come è possibile vedere nell'immagine sottostante.

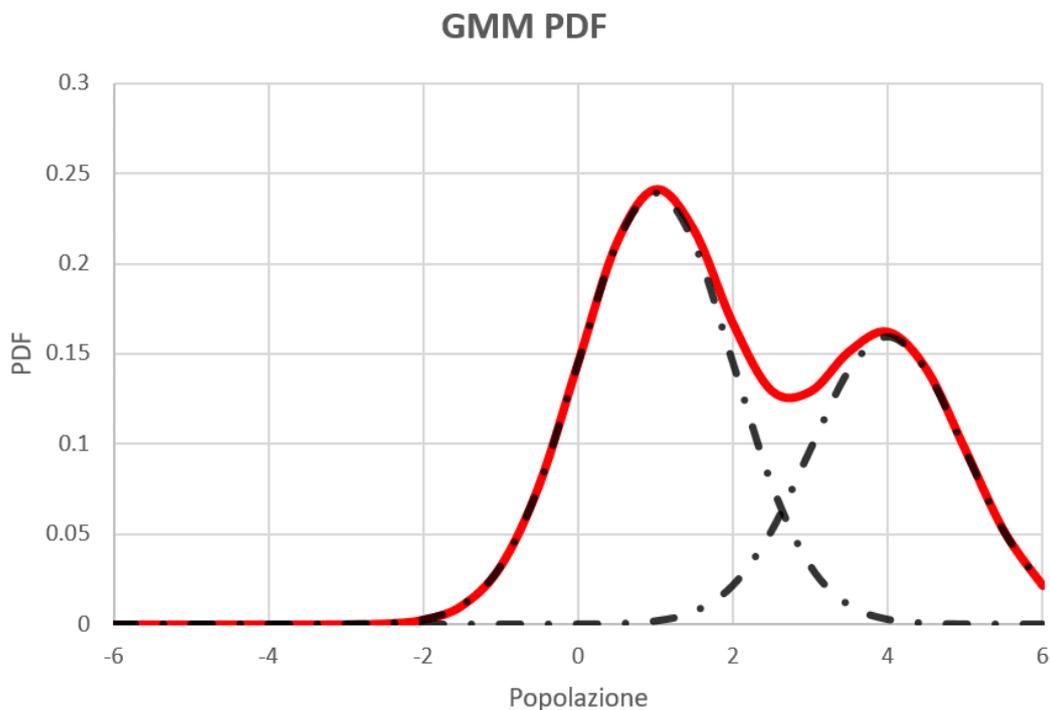


Figura 12 Gaussian Mixture Model: individui con diversa pesatura  $\pi_i$ .

Affidare maggior peso agli individui più promettenti porta quindi l'algoritmo a concentrarsi soprattutto nell'intorno di quest'ultimi, velocizzando il processo di ricerca dell'ottimo, ma, se si eccede nella pesatura, si può presentare il rischio di stazionarsi su un falso minimo. Nell'algoritmo proposto è stata operata un semplice redistribuzione lineare dei pesi andando a sottrarre quota parte del peso dagli individui selezionati peggiori.

La distribuzione dei pesi è stata così definita:

$$\pi_i = \frac{1}{n} + \left[ \frac{-2c}{n} (i) + c \right] \quad \text{con } \pi_i \geq 0$$

Dove:

- $n$  = numero di individui selezionati
- $c$  = quantità di peso redistribuita
- $i$  = indice dell'individuo;  $i = 0$  individuo migliore;  $i = n$  individuo peggiore.

## 2.4.2 VARIANZA

La matrice di varianza-covarianza, fornendo una misura della variabilità della distribuzione, permette di aumentare o diminuire l'ampiezza delle distribuzioni normali, offrendo il vantaggio, a seconda della situazione, di generare nuovi individui con caratteristiche più o meno variabili dall'individuo generatore o "madre". Questo strumento è di fondamentale importanza nel processo di ricerca dell'ottimo poiché, aumentando il valore della varianza, è possibile focalizzare l'algoritmo su un'esplorazione più ampia del dominio; mentre, diminuendone il valore, permette di affinare la precisione nella ricerca dell'ottimo. Quindi considerando  $k_{max}$  come il numero massimo di iterazioni, al fine di promuovere l'esplorazione del dominio all'inizio del processo di ottimizzazione iterativa e di promuovere la precisione verso la fine del processo, l'EDA adotta un approccio di adattamento dinamico della varianza. Definendo  $\sigma_j$  come termine diagonale di varianza della matrice di varianza-covarianza  $\Sigma$  riferito alla variabile di progetto  $j$ ,  $\sigma_j$  è stato delimitato tra un intervallo ammissibile:

$$\sigma_{j,max} = \alpha \frac{|x_j^{ls} - x_j^{li}|}{m} \quad ; \quad \sigma_{j,min} = \alpha \frac{|x_j^{ls} - x_j^{li}|}{m \cdot k_{max}}$$

Dove il parametro  $\alpha > 0$  è un iperparametro statico definito dall'utente che controlla euristicamente l'esplorazione e la precisione: aumentando  $\alpha$  si favorisce l'esplorazione, al contrario diminuendolo, si favorisce la ricerca.  $\sigma_{j,max}$  e  $\sigma_{j,min}$  rappresentano gli estremi della variazione dinamica di  $\sigma_j$  durante il processo di calcolo [7].

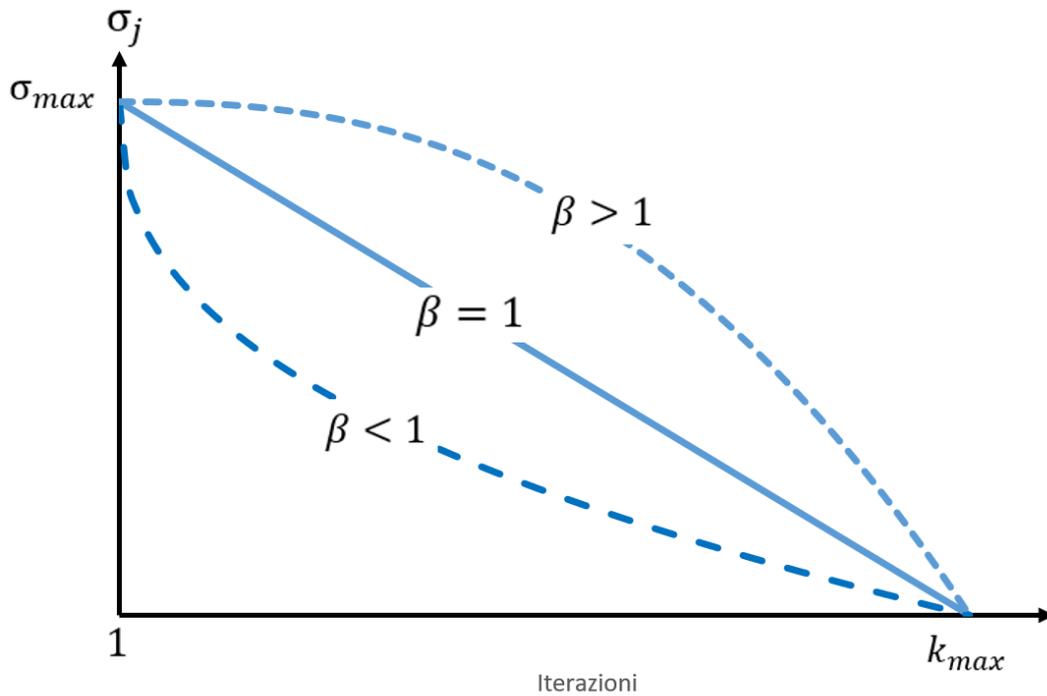


Figura 13 Variazione dinamica di  $\sigma$  in funzione di  $\beta$

Per controllare ulteriormente questi aspetti, è stata definita una legge di variazione dinamica continua decrescente:

$$\sigma_j(k) = \sigma_{j,max} - \frac{\sigma_{j,max} - \sigma_{j,min}}{\beta} k^\beta$$

dove il parametro  $\beta > 0$  è un ulteriore iperparametro statico definito dall'utente che regola euristicamente l'andamento della legge decrescente durante l'iterazione, il quale può essere impostato in tre differenti modi:

- Andamento convesso per  $0 < \beta < 1$
- Andamento lineare per  $\beta = 1$
- Andamento concavo per  $\beta > 1$

### 2.4.3 CODICE

Per la scrittura del codice riguardante la creazione del modello probabilistico è stato utilizzato in linguaggio Matlab il comando `“gmdistribution(mu,sigma,p)”` dove “mu” sono le caratteristiche dell’individuo, “sigma” la matrice di varianza-covarianza e “p” il peso del individuo. È stato poi possibile estrarre una nuova generazione di individui dalla distribuzione grazie all’utilizzo del comando `“random(gm,n)”` dove “gm” è il modello probabilistico e “n” è il numero di individui da generare.

Esempio in linguaggio Matlab:

```
gm = gmdistribution(mu,sigma,p)
nuovi_individui = random(gm,n)
```

Per quanto riguarda l’implementazione del codice in linguaggio Python, è stato possibile creare il modello probabilistico grazie all’utilizzo della libreria Numpy. Non essendo presente un comando ad hoc per la generazione del Gaussian Mixture Model come in Matlab, è stata utilizzata una sequenza di comandi che permettesse una scelta stocastica delle gaussiane definite dagli individui generatori. Sono stati utilizzati quindi i comandi `“random.choice(mat_ind, n_ind, p)”` per una scelta stocastica del modello Gaussiano da cui generare il nuovo individuo e `“random.multivariate_normal( mat_ind, cov, ind_estr)”` per l’estrazione random da distribuzione normale del nuovo individuo.

Esempio codice Python:

```
indice_individuo= numpy.random.choice( mat_ind, n_ind, p)
for i in range(n_ind):
    nuovi_individui[i,:]=numpy.random.multivariate_normal(mat_ind[indice_individuo],
cov, ind_estr)
```

Dove “mat\_ind” è la matrice contenente le caratteristiche degli individui selezionati, “n\_ind” il numero di individui selezionati, “cov” la matrice di varianza-covarianza, “ind\_estr” il numero di individui da estrarre per una singola gaussiana.

I codici proposti, per la generazione modello probabilistico, rivelano però una criticità legata ai vincoli imposti dal dominio della funzione, la quale si manifesta nel momento in cui il nostro

campione si addensa sul bordo del dominio, portando a generare candidati fuori dai vincoli imposti. Essendo molto difficile annullare la probabilità generata fuori dal dominio, sia per costo computazionale che per quanto riguarda la stesura del codice, sono stati implementati e confrontati i seguenti metodi:

- Metodo della penalità: Ad ogni individuo fuori dal dominio è stata sommata alla propria funzione obiettivo una penalità proporzionale alla distanza dal bordo del dominio.
- Metodo dello specchio: Al fine di far rientrare gli individui generati fuori dal dominio nei limiti di quest'ultimo, sono state specchiate le coordinate del candidato in funzione del bordo del dominio.
- Metodo della traslazione: Tale metodo trasla molto semplicemente il nostro candidato sul bordo più vicino del dominio.

È stato possibile vedere che tra i vari metodi, il metodo della traslazione, anche se meno accurato, è stato il più efficace poiché richiede poco costo computazionale ed è facilmente implementabile.

### 3 PROBLEMI NON VINCOLATI

La verifica dell'affidabilità, dell'efficienza e della validazione degli algoritmi di ottimizzazione passa attraverso la risoluzione di test di benchmark standard o di funzioni di test tratte dalla letteratura. Per qualsiasi nuovo algoritmo di ottimizzazione, è essenziale convalidare le sue prestazioni e confrontarle con quelle di altri algoritmi esistenti su un buon insieme di funzioni di test.

Tuttavia, va notato che l'efficacia di un algoritmo rispetto ad altri non può essere misurata semplicemente in base ai problemi che risolve, se l'insieme dei problemi è troppo specializzato e privo di proprietà diverse. Perciò, per valutare un algoritmo, è necessario identificare il tipo di problema in cui si comporta meglio rispetto agli altri cosa che aiuta a caratterizzare il tipo di problema per i quali un algoritmo è più adatto.

Nella fase preliminare di test dell'algoritmo sono stati affrontati problemi artificiali che possono essere utilizzati per valutare il comportamento di un algoritmo in situazioni talvolta diverse e difficili. I problemi artificiali possono includere un singolo minimo globale, uno o più minimi globali in presenza di molti minimi locali, valli lunghe e strette, effetti di spazio nullo e superfici piane. Questi problemi possono essere facilmente manipolati e modificati per testare gli algoritmi in diversi scenari [8].

Sono stati settati un insieme di test di benchmark su funzioni con diverse proprietà, come la modalità e la separabilità.

### 3.1 MODALITÀ

Una funzione con più di un ottimo locale è detta multimodale; queste funzioni sono utilizzate per testare la capacità di un algoritmo di uscire da un minimo locale. Se il processo di esplorazione del dominio è mal progettato si può incorrere in uno stazionamento dell'algoritmo su di un minimo locale per questo motivo le funzioni multimodali con molti minimi locali sono tra le classi di problemi più difficile per molti algoritmi [8].

### 3.2 SEPARABILITÀ

La separabilità è una misura della difficoltà delle diverse funzioni di riferimento. In generale, le funzioni separabili sono relativamente facili da risolvere, se confrontate con la loro controparte inseparabile, perché ogni variabile di una funzione è indipendente dalle altre variabili. Se tutti i parametri o variabili sono indipendenti, è possibile eseguire una sequenza di  $n$  processi di ottimizzazione indipendenti.

Di conseguenza, ogni variabile o parametro del progetto può essere ottimizzato in modo indipendente. La condizione generale di separabilità per verificare se la funzione è facile da ottimizzare o meno è data da

$$\frac{\partial f(\bar{x})}{\partial x_i} = g(x_i)h(\bar{x})$$

dove  $g(x_i)$  indica una funzione qualsiasi di  $x_i$  e  $h(\bar{x})$  una funzione qualsiasi di  $(\bar{x})$ . Se questa condizione è soddisfatta, la funzione è detta parzialmente separabile e facilmente ottimizzabili, perché le soluzioni per ogni  $x_i$  possono essere ottenute indipendentemente da tutti gli altri parametri [8].

### 3.3 DEFINIZIONE DEL PROBLEMA

L'obiettivo di un qualsiasi problema di ottimizzazione è di trovare le migliori soluzioni possibili  $x$  da un insieme  $X$  le quali rispettano un insieme di criteri  $F = \{f_1, f_2, \dots, f_n\}$ . L'insieme di questi criteri è chiamato funzione obiettivo, espresso sotto forma di funzioni matematiche. Una funzione obiettivo è una funzione matematica  $f: D \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$  soggetta a vincoli aggiuntivi, dove l'insieme  $D$  è il dominio della funzione. Nel caso dell'ottimizzazione di un singolo criterio  $f$ , un ottimo è il suo massimo o il suo minimo. I problemi di ottimizzazione globale sono spesso definiti come problemi di minimizzazione, tuttavia, questi problemi possono essere facilmente convertiti in problemi di massimizzazione invertendo il segno della funzione obiettivo.

Un problema di ottimizzazione generico può essere quindi così definito:

$$\text{minimize } f(x)$$

La soluzione ottimale di un problema di ottimizzazione può essere un insieme di  $x^* \in D$  di tutti i punti ottimali in  $D$ , piuttosto che un singolo valore minimo o massimo in alcuni casi. Le soluzioni ottimali possono essere molteplici, o addirittura infinite, a seconda del dominio dello spazio di ricerca [8].

### 3.4 TEST DI BENCHMARK

Sono qui esposti i problemi di ottimizzazione non vincolati con i quali sono stati effettuati test sull'algoritmo proposto. Sono state scelte funzioni bidimensionali per avere un maggior controllo sull'evolversi del modello probabilistico.

Poiché l'approccio proposto consente all'utente di variare gli iperparametri  $\alpha$  e  $\beta$  per governare l'adattamento dinamico della matrice di covarianza durante il processo di ottimizzazione, i problemi di riferimento sono stati testati con diversi con diversi valori di  $\beta$ , lasciando  $\alpha$  fisso, esplorando così gli effetti di una matrice di covarianza con un andamento concavo, lineare e convesso. A scopo di confronto, le prestazioni dell'EDA sono state confrontate con altri EA diversi per esplorare l'affidabilità e l'efficacia dell'ottimizzazione. Sono stati selezionati i seguenti algoritmi come EA alternativi di implementazione completamente indipendente per una valutazione oggettiva della robustezza, dell'affidabilità e delle prestazioni dell'attuale EDA:

- Funzione sferica: confronto con il GA dall'implementazione della libreria open-source PyGAD di Python.
- Funzione di Himmelblau: confronto con il GA implementato da ChenNanyang, L. P. Khoo and C. H. [9]
- Funzione di Ackley: confronto con il GA dall'implementazione della libreria open-source PyGAD di Python.
- Funzione di Rastrigin: confronto con il GA dall'implementazione standard di Matlab e il PSO, nella variante del PSO multi-strategia potenziato (PSO-MS) [7].

### 3.4.1 FUNZIONE SFERICA

La funzione sferica rappresenta un semplice problema di ottimizzazione non vincolata poiché è una funzione a variabili separabili e non è modale. Il problema è così definito:

$$f(x) = \sum_{i=1}^n x_i^2 \quad -10 < x < 10$$

La funzione presenta un minimo assoluto per  $f(x_1, \dots, x_n) = f(0, \dots, 0) = 0$

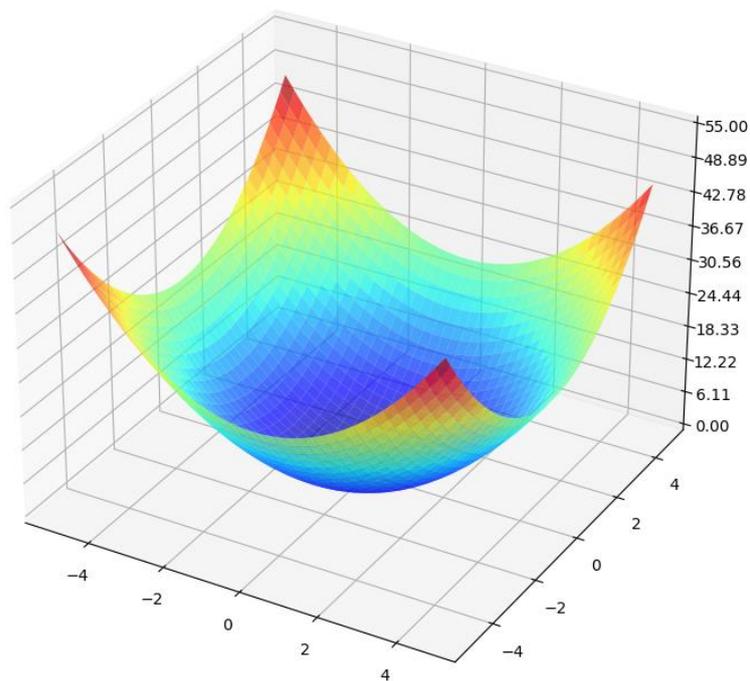


Figura 14 Rappresentazione bidimensionale della funzione sferica.

Sono stati eseguiti 50 test indipendenti con una popolazione di partenza di 50 individui e un numero massimo di iterazioni pari a 200.

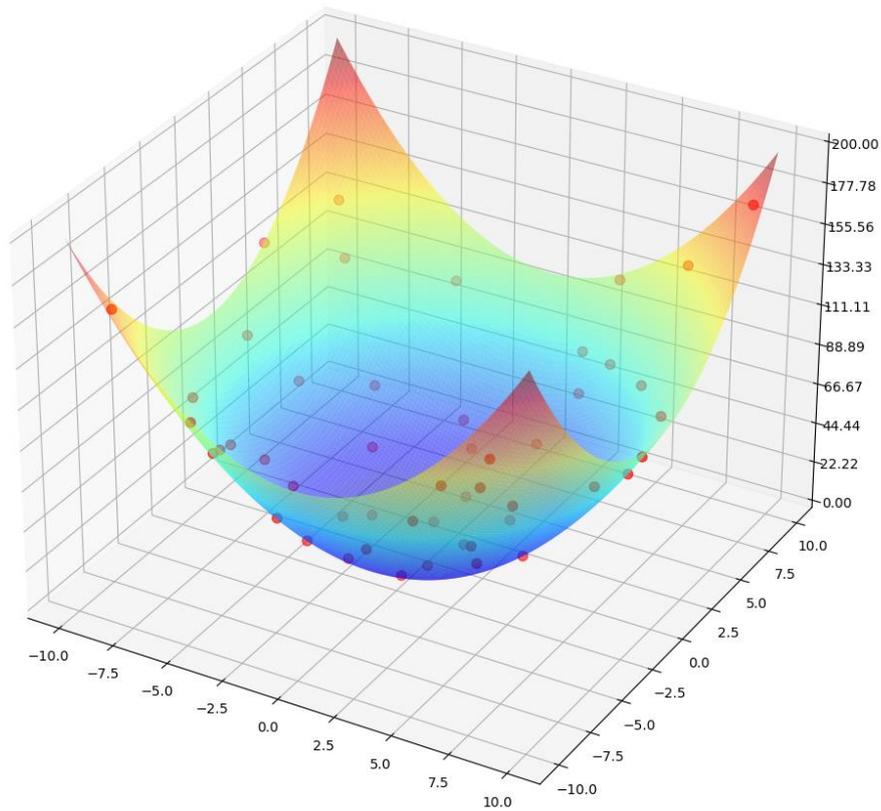


Figura 15 In rosso la prima generazione di individui.

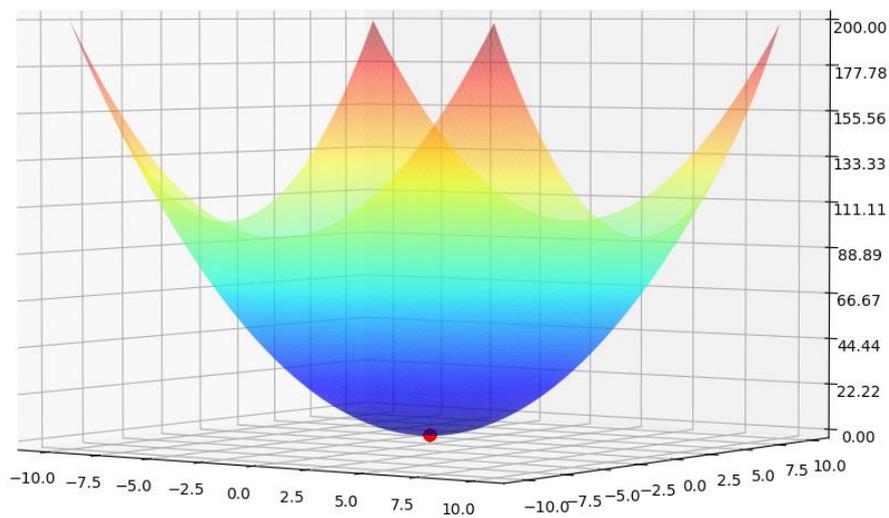


Figura 16 In rosso la miglior soluzione dopo la duecentesima iterazione.

Per avere un confronto quanto più idoneo possibile è stato considerato un valore di  $\alpha=0.01$  in modo tale che l'accuratezza dei risultati rispecchiasse quella del GA. I valori del parametro  $\beta$  sono stati impostati variando da un minimo di 0.5 a un massimo di 2.5 con un incremento di 0.5.

<b>Obj. Fun.</b>	<b>Real Optimum</b>	<b>GA</b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=0.5</math></b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=1</math></b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=1.5</math></b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=2</math></b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=2.5</math></b>
best	0.0	3.83408E-10	3.7399E-09	2.8069E-10	2.12853E-08	1.75817E-08	2.15204E-08
worse	-	2.45567E-05	1.87645E-06	4.52063E-06	7.26581E-06	2.9435E-06	4.99927E-06
mean	-	3.62018E-06	4.52116E-07	6.75968E-07	1.11561E-06	1.12013E-06	1.30317E-06
std. dev.	-	4.99975E-06	4.11263E-07	8.55436E-07	1.21495E-06	8.02826E-07	1.18116E-06

*Risultati algoritmo EDA e confronto*

Dai dati si evince un ottimo comportamento globale dell'EDA mantenendo una media allineata e non inferiore ai risultati del GA e con una deviazione standard minore. Si evince che le impostazioni per la ricerca del miglior punto di ottimo sono per  $\beta=1$  mostrando una compattezza del modello con una deviazione standard molto piccola.

### 3.4.2 FUNZIONE DI HIMMELBLAU

La funzione di Himmelblau è una funzione multimodale a variabili separabili che presenta quattro minimi assoluti. Il problema è così definito:

$$f(x) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 + 0.1 \cdot |(x - 3)^2 + (y - 2)^2| \quad -6 \leq x, y \leq 6$$

La funzione presenta un minimo assoluto per  $f(x, y) = f(3.0, 2.0) = 0$

La funzione di Himmelblau è caratterizzata per avere altri tre minimi locali molto vicino all'ottimo per i valori di  $(-2.805118, 3.131312)$  ,  $(-3.779310, -3.283186)$  e  $(3.584428, -1.848126)$ .

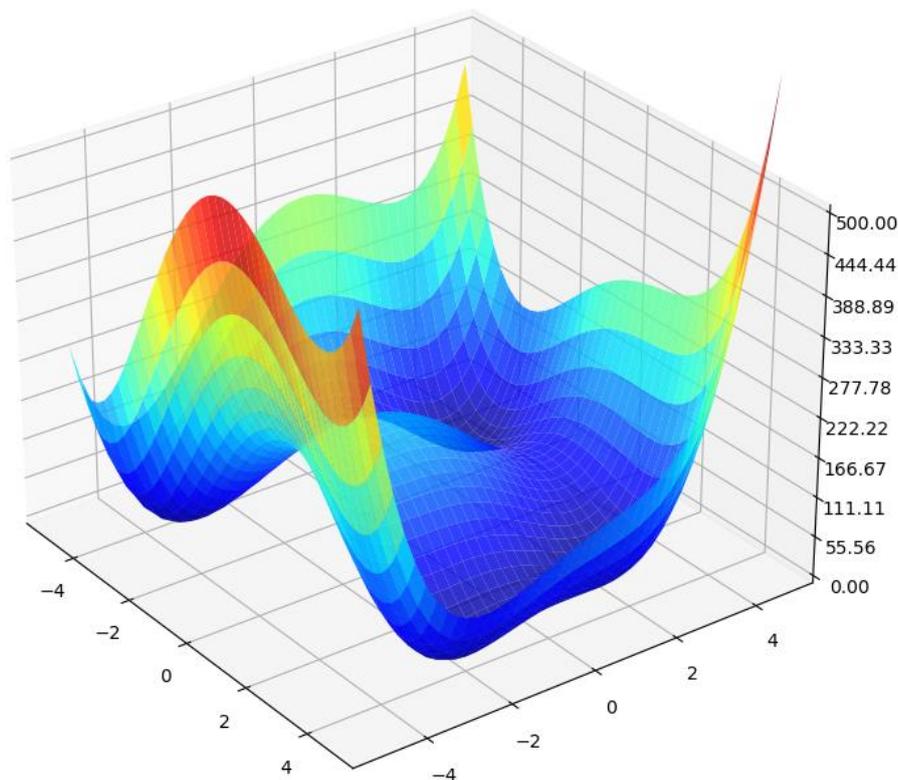


Figura 17 Rappresentazione grafica della funzione di Himmelblau.

Sono state condotte 6000 iterazioni con una popolazione di partenza di 50 individui e riportato un confronto sui dati restituiti a determinate iterazioni dal GA [9]. In questo caso è stata fissato un valore di  $\alpha=1$  e  $\beta=0.5$ .

Generation	GA [9]			EDA $\alpha=0.1$ $\beta=0.5$		
	X	Y	Obj. Fun.	X	Y	Obj. Fun.
1	-2.973	2.468	3.6832	2.590	2.796	13.9335
1	2.320	5.322	1.2379	2.861	1.840	1.5159
8	2.952	2.468	0.0570	2.880	2.088	0.4378
97	2.897	2.430	0.0504	3.006	2.022	0.0126
128	2.957	2.064	0.0372	3.006	2.022	0.0126
195	2.927	2.395	0.0176	2.996	2.023	0.0080
362	2.961	2.250	0.0068	2.996	2.023	0.0080
979	3.032	1.854	0.0043	2.995	2.003	0.0007
1608	3.024	1.850	0.0023	3.003	1.996	0.0004
3588	3.008	1.987	0.0014	3.003	1.996	0.0004
5825	3.001	1.994	0.0000	3.001	2.003	0.0000

*Risultati algoritmo EDA e confronto*

Dai risultati si può notare come l'EDA arrivi molto più velocemente a convergere sull'ottimo della funzione oltre a fornire un modello più accurato nella definizione dei dati.

### 3.4.3 FUNZIONE DI ACKLEY

La funzione di Ackley è una funzione multimodale a variabili non separabili. Il problema è così definito:

$$f(x) = -20 \cdot \exp \left[ -0.2 \sqrt{0.5(x^2 + y^2)} \right] + \exp \left[ -0.5(\cos 2\pi x + 2) \right] + e + 20$$
$$-5 \leq x, y \leq 5$$

La funzione presenta un minimo assoluto per  $f(x, y) = f(0, 0) = 0$

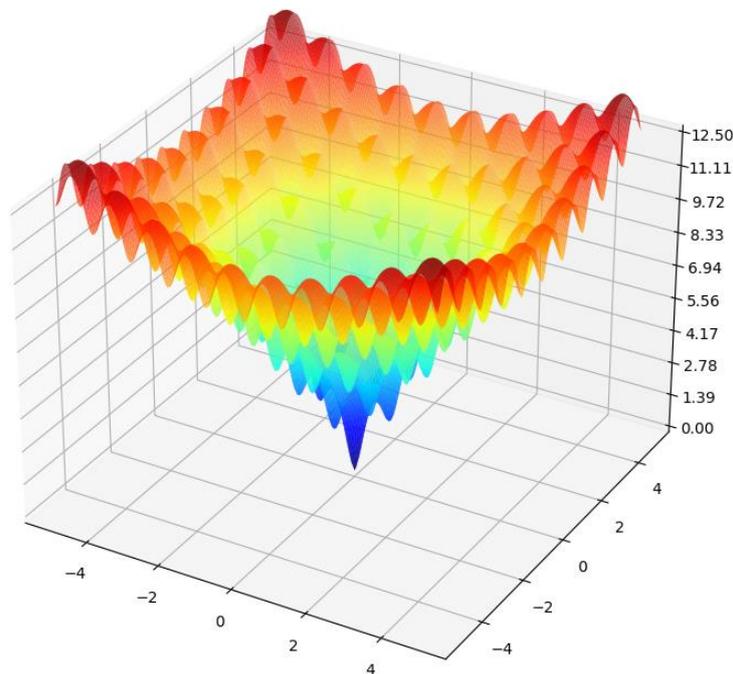


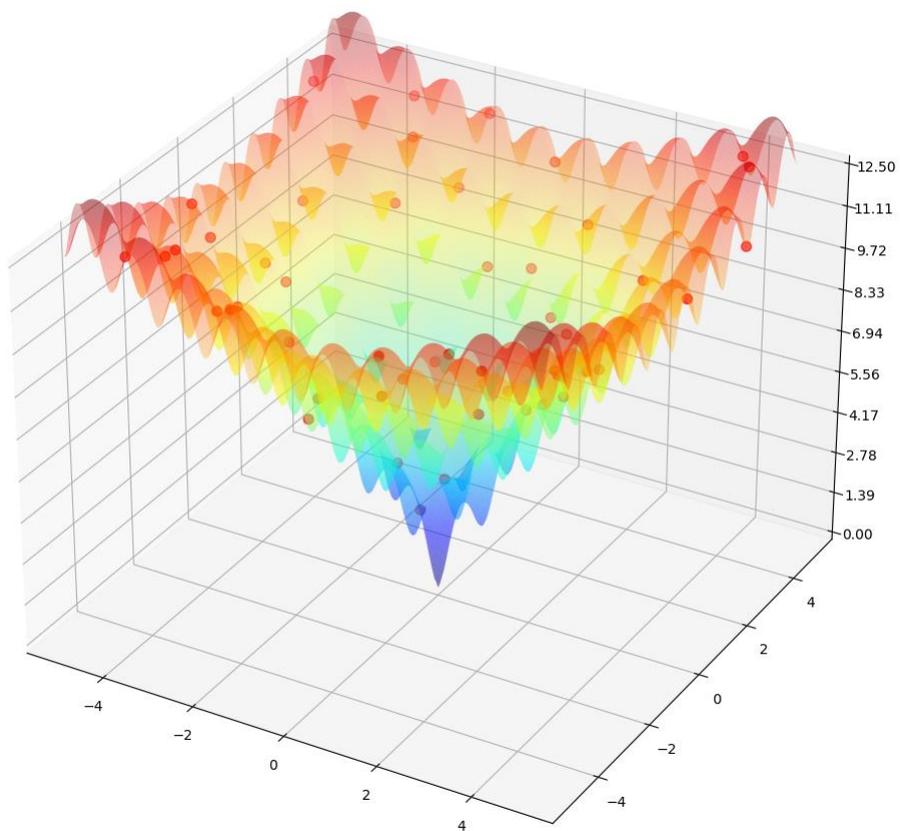
Figura 18 Rappresentazione grafica della funzione di Ackley.

Per avere valori quanto più omogenei possibili e fare un giusto confronto con l'algoritmo GA sono state diverse considerazioni per la soluzione del problema di Ackley, in particolare considerando due moltiplicatori della varianza:

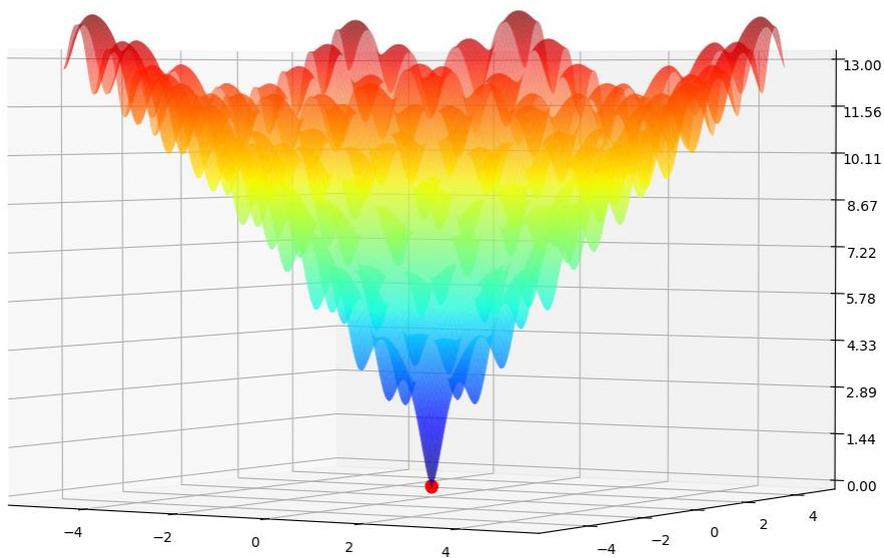
$$\sigma_{j,max} = \alpha_{max} \frac{|x_j^{ls} - x_j^{li}|}{m} \quad ; \quad \sigma_{j,min} = \alpha_{min} \frac{|x_j^{ls} - x_j^{li}|}{m \cdot k_{max}}$$

Cambiando in tal modo le impostazioni di  $\alpha$  ci ha permesso di ampliare il range di variazione della varianza  $\sigma_j$  così da avere un'esplorazione maggiore nella fase iniziale e una precisione

migliore nella fase finale. Sono stati eseguiti 50 test indipendenti con una popolazione di partenza di 50 individui e un numero massimo di iterazioni pari a 200.



*Figura 19 In rosso la prima generazione di individui.*



*Figura 20 In rosso la miglior soluzione dopo la duecentesima iterazione.*

I valori del parametro  $\beta$  sono stati impostati variando da un minimo di 0.5 a un massimo di 2.5 con un incremento di 0.5 mentre sono stati fissati i valori  $\alpha$  pari a  $\alpha = 0.01$ .

<b>Obj. Fun.</b>	<b>Real Optimum</b>	<b>GA</b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=0.5</math></b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=1</math></b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=1.5</math></b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=2</math></b>	<b>EDA <math>\alpha=0.01</math> <math>\beta=2.5</math></b>
best	0.0	0.000637801	0.000635892	0.000541332	0.000156464	0.001242332	0.0005636
worse	-	0.014618798	0.007143016	0.008683318	0.012184402	0.018242169	0.0170564
mean	-	0.005057802	0.003487145	0.004691796	0.005036758	0.006523065	0.00629228
std. dev.	-	0.003676333	0.001874306	0.001969694	0.002647381	0.003685919	0.003835893

*Risultati algoritmo EDA e confronto*

Dalla tabella si può vedere come i risultati siano in linea con quelli del GA. La condizione migliore per la soluzione della funzione di Ackley è considerando un  $\beta=0.5$ , infatti, l'EDA fornisce un modello più compatto se confrontato con il GA avendo una deviazione standard inferiore e con una media dei valori migliore.

### 3.4.4 FUNZIONE DI RASTRIGIN

La funzione di Rastrigin è una funzione multimodale a variabili separabili. Il problema è così definito:

$$f(x) = \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i) + 10 \quad -5.12 < x < 5.12$$

La funzione presenta un minimo assoluto per  $f(x_1, \dots, x_n) = f(0, \dots, 0) = 0$

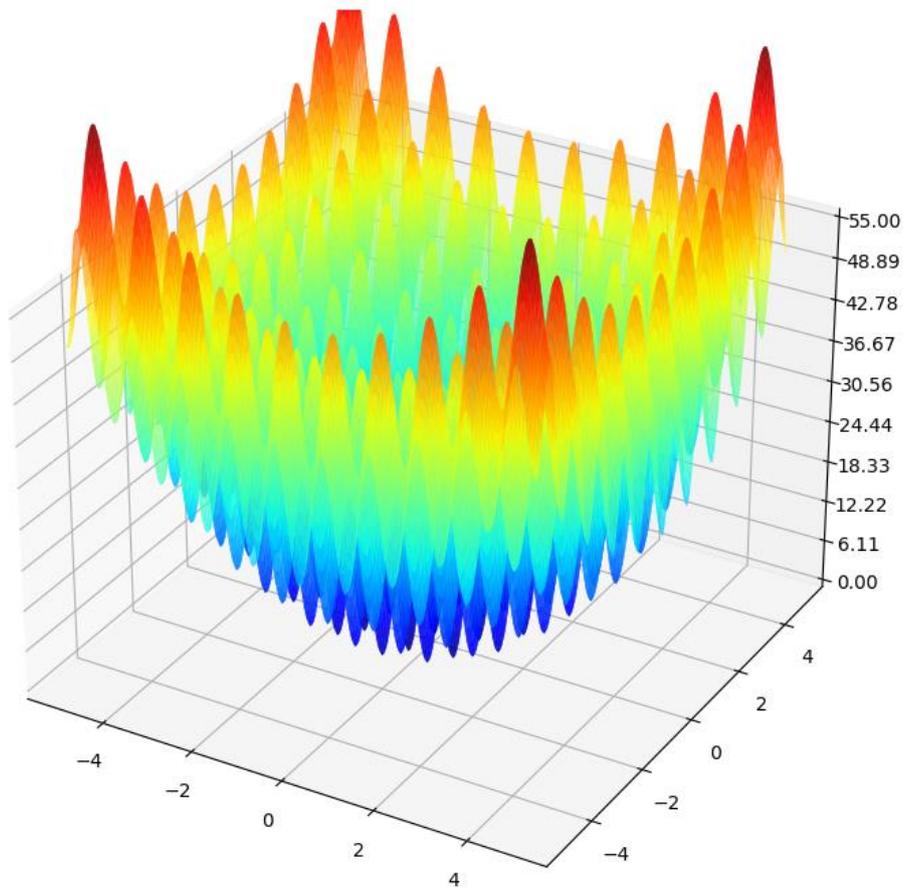


Figura 21 Rappresentazione bidimensionale della funzione di Rastrigin.

Per la risoluzione del problema sono stati lanciati 50 analisi indipendenti con una popolazione di 50 individui e il numero massimo di iterazioni  $k_{max} = 500$ . I valori del parametro  $\beta$  sono stati impostati variando da un minimo di 0.5 a un massimo di 2.5 con un incremento di 0.5 mentre è stato fissato il valore  $\alpha = 1$ .

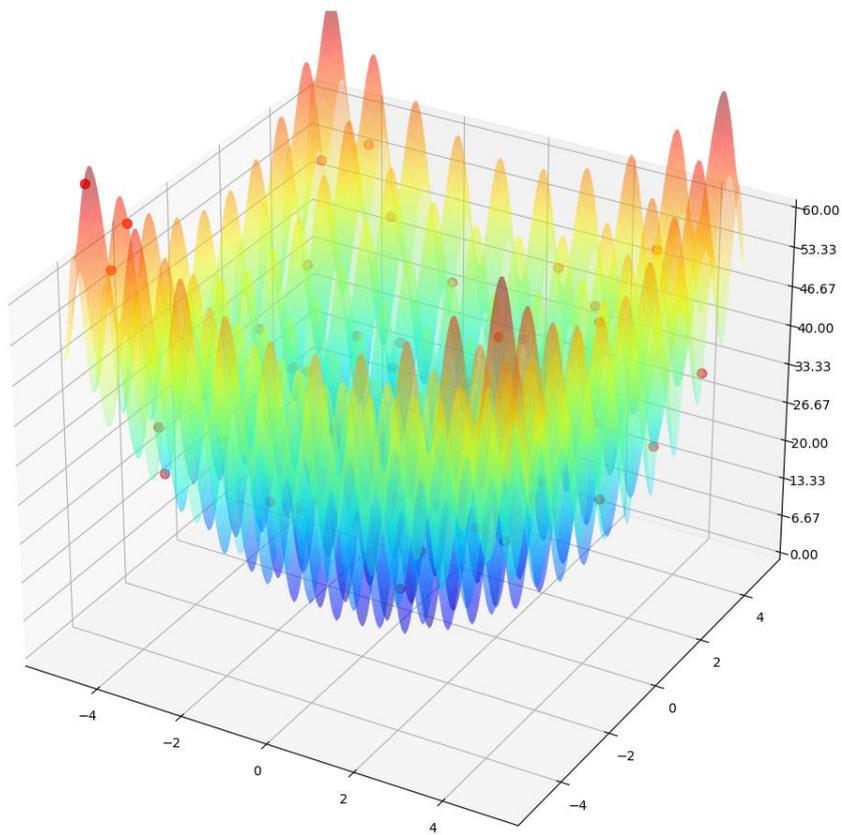


Figura 22 In rosso la prima generazione di individui.

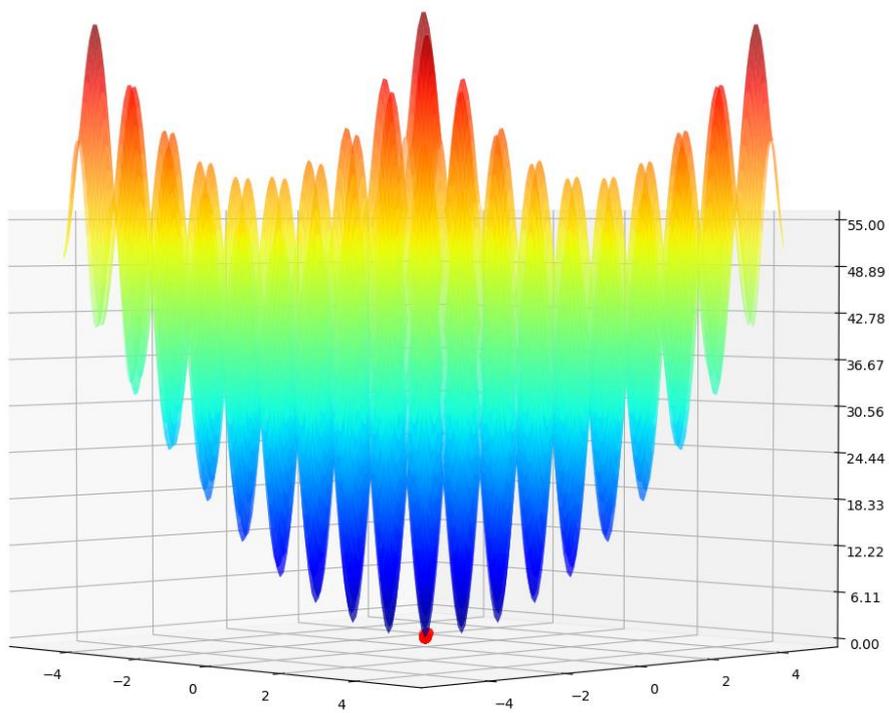


Figura 23 In rosso l'ottimo della funzione calcolato dall'EDA.

Obj. Fun.	Real Optimum	PSO [7]	GA [7]	EDA	EDA	EDA	EDA	EDA
				$\alpha=0.01$ $\beta=0.5$	$\alpha=0.01$ $\beta=1$	$\alpha=0.01$ $\beta=1.5$	$\alpha=0.01$ $\beta=2$	$\alpha=0.01$ $\beta=2.5$
best	0	0	3.73E-10	6.13E-07	7.29E-07	5.01E-07	5.21E-07	1.31E-06
worse	-	0	3.98	4.23E-05	4.91E-05	0.95	1.03E-04	1.34E-04
mean	-	0	0.46	7.07E-06	1.55E-05	0.02	2.70E-05	3.29E-05
std.dev.	-	0	0.76	6.23E-06	1.36E-05	0.13	2.40E-05	3.02E-05

*Risultati algoritmo EDA e confronto*

I risultati dell'EDA appaiono molto promettenti, effettivamente vicini all'optimum globale e con una variabilità molto bassa rispetto al GA. Il PSO-MS appare molto robusto poiché fornisce, per tutte le 50 esecuzioni indipendenti, la soluzione ottimale globale senza alcuna variazione. L'effetto di  $\beta$  sui risultati ottimali appare evidente, anche se in questo caso è molto basso, fornendo in tutti i casi, tranne  $\beta = 1,5$ , risultati notevoli, soprattutto rispetto ai GA standard. [7]

## 4 PROBLEMI VINCOLATI

A seguito dei buoni risultati dei test sui problemi non vincolati, sono stati implementati dei test di benchmark su problemi vincolati, in modo che si potesse analizzare il comportamento dell'algoritmo proposto su problemi quanto più simili ai problemi strutturali reali.

L'introduzione dei vincoli porta a un incremento consistente della complessità del problema, generando in primo luogo una distinzione sostanziale tra gli individui "feasible", i quali rispettano i vincoli imposti, dagli individui "unfeasible", i quali non rispettano i vincoli imposti. Tale distinzione ci obbliga a dover ridefinire in modo quanto più possibile congruente la funzione obiettivo affidando una penalità agli individui "unfeasible", senza però del tutto escluderli nell'influenzare la creazione del modello probabilistico. Per fare ciò è stata definita una funzione di penalità iterativa proporzionale alla distanza degli individui "unfeasible", dagli individui "feasible" più vicini.

Sono stati impostati due tipi di benchmark uno di tipo artificiale sul problema di Sickle e un problema strutturale sul problema denominato Truss 10.

### 4.1 DEFINIZIONE DEL PROBLEMA

Un problema di ottimizzazione vincolato è generalmente così definito:

$$\text{Maximize or minimize } f(\bar{x})$$

Soggetta a:

$$g_j(\bar{x}) \leq 0 \quad j = 1, 2, \dots, J$$

$$h_k(\bar{x}) = 0 \quad k = 1, 2, \dots, K$$

$$X_i^l \leq x_i \leq X_i^u \quad i = 1, 2, \dots, I$$

Il problema riportato sopra presenta  $N$  variabili ( $\bar{x}$  è un vettore di dimensioni  $N$ ),  $J$  vincoli di disuguaglianza e  $K$  vincoli di uguaglianza. La funzione  $f(\bar{x})$  è la funzione obiettivo,  $g_j(\bar{x})$  è il  $j$ -esimo vincolo di disuguaglianza e  $h_k(\bar{x})$  è il  $k$ -esimo vincolo di uguaglianza. La  $i$ -esima variabile può mutare nel dominio compreso tra  $X_i^l$  e  $X_i^u$ .

## 4.2 DEFINIZIONE DELLA PENALITÀ

Nella risoluzione dei problemi vincolati sorge il problema di come trattare tutti quegli individui che non rispettano i vincoli imposti dal problema e vanno a generare un sottodominio di individui “unfeasible”.

La metodologia più semplice per trattare gli individui che non rispettano i vincoli è quella di applicare una penalità  $P(x)$  costante alla funzione obiettivo:

$$F^a(x) = F(x) \quad \text{se } x \in \tilde{F}$$

$$F^a(x) = F(x) + P$$

Dove  $F^a(x)$  è la funzione obiettivo aumentata della penalità,  $\tilde{F}$  è il sottodominio degli individui “feasible” e  $P$  è la funzione penalità eguagliata a una costante.

Essendo troppo approssimativo l'utilizzo del metodo precedente sono stati definiti dei metodi che mettessero in relazione la violazione dei vincoli con la funzione di penalità andando così a definire una funzione  $P(x)$  variabile:

$$F^a(x) = F(x) \quad \text{se } x \in \tilde{F}$$

$$F^a(x) = F(x) + P(x)$$

$$P(x) = \sum_{j=1}^J (\lambda_G)_j [G_j(x)]^\beta + \sum_{k=1}^K (\lambda_H)_k [H_k(x)]^\beta$$

Dove:

$$G_j(x) = \max[0, g_j(x)]$$

$$H_k(x) = \text{abs}[h_k(x)]$$

$G_j(x)$  e  $H_k(x)$  rappresentano rispettivamente il grado di violazione dei vincoli di uguaglianza e disuguaglianza mentre  $(\lambda_G)_j$ ,  $(\lambda_H)_k$  e  $\beta$  sono delle costanti. Nella maggior parte dei casi vengono utilizzati gli stessi valori per  $(\lambda_G)_j$ ,  $(\lambda_H)_k$  mentre per  $\beta$  viene impostato un valore tra 1 e 2. Il grado di penalità viene corretto impostando i valori di  $(\lambda_G)_j$ ,  $(\lambda_H)_k$  i quali semplificano sensibilmente l'operazione ma in taluni casi non è così ovvio affidarne un valore. [note]

Partendo dall'equazione precedente è stato creato un metodo adattivo di penalizzazione  $P(x)$  tale che potesse mettere in relazione tra loro il grado di violazione e la distanza degli individui dalla frontiera della zona “feasible”  $\tilde{F}$ , nella quale si trovano tutte le soluzioni che rispettano i vincoli. Per far ciò è stata impostata la funzione penalità  $P(x)$  come un gradiente di penalizzazione in funzione della distanza dalla zona “feasible”  $\tilde{F}$ . Non essendo a conoscenza in maniera esatta della frontiera della zona  $\tilde{F}$  è stato quindi messa in relazione la funzione penalità  $P(x)$  con la distanza euclidea dall'individuo “feasible” più vicino al corrispettivo individuo “unfeasible”. Questo però vale dal momento che nella nostra popolazione è presente almeno un individuo “feasible” su cui basarci per definire la nostra penalità. Per tale ragione è stato necessario creare un ulteriore definizione della funzione penalità  $P(x)$  la quale andasse ad affidare una valutazione in funzione della distanza dal bordo della zona “feasible” nella condizione che nella popolazione generata ci fossero solo individui che non rispettano i vincoli imposti.

Così viene definita la valutazione degli individui “unfeasible”:

se  $\tilde{X} = \emptyset$

$$F^a(x) = n_{g,h} \cdot \left( \sum_{j=1}^J g_j(x) + \sum_{k=1}^K h_k(x) \right)$$

se  $\tilde{X} \neq \emptyset$

$$F^a(x) = F(\tilde{x}_{near}) + d_{\tilde{x}_{near}}^2 \cdot n_{g,h} \cdot \left( \sum_{j=1}^J g_j(x) + \sum_{k=1}^K h_k(x) \right)$$

$$\text{Con } \begin{cases} g_j(x) = 0 & \text{se } g_j(x) \leq 0 \\ h_k(x) = 0 & \text{se } h_k(x) = 0 \end{cases}$$

$\tilde{X}$  rappresenta l'insieme degli individui che rispettano i vincoli presenti nella popolazione da valutare,  $n_{g,h}$  corrisponde al numero dei vincoli che sono stati violati,  $F(\tilde{x}_{near})$  è il valore della funzione obiettivo dell'individuo più vicino e  $d_{\tilde{x}_{near}}$  è la distanza euclidea tra l'individuo “feasible” più vicino e l'individuo “unfeasible” in stato di valutazione.

Definendo in tale modo la funzione  $F^a(x)$ , si crea una prima fase dove avviene una ricerca della zona “feasible”, poiché gli individui che vengono privilegiati sono quelli più vicino alla frontiera, e una seconda fase di penalizzazione adattiva.

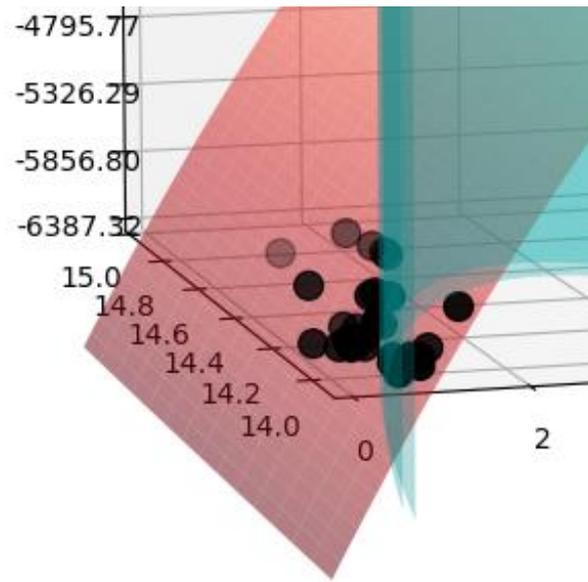


Figura 24 Valutazione funzione penalty su problema di Sickle: in rosso il dominio della funzione obiettivo, in ciano la frontiera dei vincoli imposti dal problema e in nero gli individui "unfeasible" valutati con il metodo di penalizzazione adattivo proposto.

### 4.3 TEST DI BENCHMARK

Sono stati risolti due tipi di test di benchmark, uno analitico e uno strutturale, in modo tale da avvicinarci quanto più possibile all'applicazione dell'algoritmo proposto in condizioni quanto più reali possibili. Dopodiché i risultati sono stati confrontati con algoritmi già presenti in commercio.

#### 4.3.1 PROBLEMA DI SICKLE

Il problema di Sickle, detto anche funzione Banana, è un problema di ottimizzazione analitico. La funzione obiettivo è definita come un piano inclinato e la zona "feasible" si trova racchiusa tra due parabole, le quali vanno a identificare il punto di ottimo nell'intersezione di quest'ultime. Il problema è così definito:

$$x \in \Omega$$

$$\text{Min } f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$$

$$\text{s. t. } g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$

$$g_2(x) = (x_1 - 6)^2 - (x_2 - 5)^2 - 82.81 \leq 0$$

Dove  $\Omega$  è il dominio del problema per le due variabili  $x_1 \in [13; 100]$  e  $x_2 \in [0; 100]$ .  $g_1(x)$  e  $g_2(x)$  sono i due vincoli di disuguaglianza del problema di forma parabolica che creano una piccola zona "feasible". Il punto di ottimo si colloca per  $x^* = [14.095; 0.84296]$  dove  $f(x^*) = -6961.81388$ . Sono stati condotti 50 test indipendenti con una popolazione di partenza di 50 individui e un numero massimo di iterazioni pari a 500.



Figura 25 Problema di Sickle: in rosso il dominio della funzione obiettivo, in nero la rappresentazione delle funzioni di vincolo che racchiudono la zona "feasible".

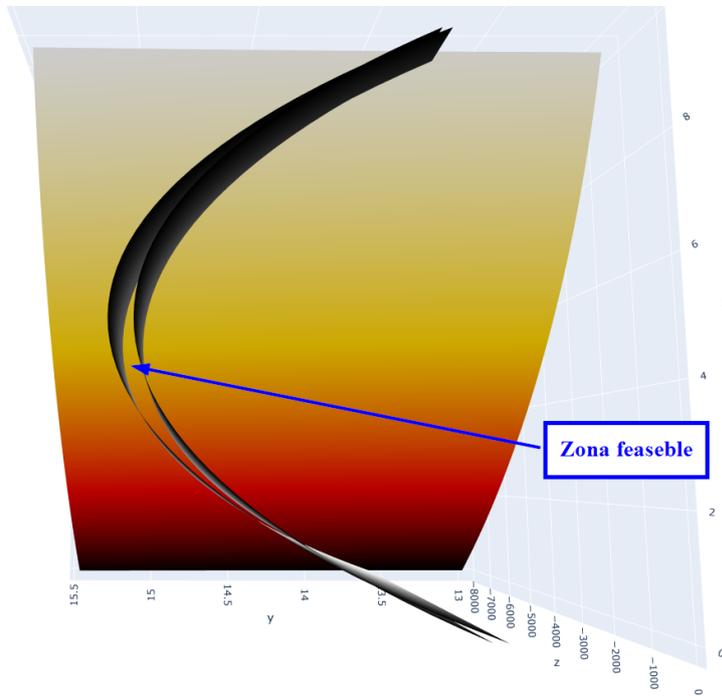


Figura 26 Problema di Sickle: dettaglio zona "feasible".

Obj. Fun.	Real Optimum	PSO [7]	GA [7]	EDA $\alpha=0.01$ $\beta=0.5$	EDA $\alpha=0.01$ $\beta=1$	EDA $\alpha=0.01$ $\beta=1.5$	EDA $\alpha=0.01$ $\beta=2$	EDA $\alpha=0.01$ $\beta=2.5$
best	6961.81388	-6961.8	9.00E+03	6.96E+03	6.96E+03	6.95E+03	6.96E+03	6.96E+03
worse	-	-6958.4	-1611.7	6.94E+03	6.93E+03	-6937.86	6.93E+03	6.92E+03
mean	-	-6960.7	-7070.8	6.95E+03	6.95E+03	-6946.49	6.95E+03	6.95E+03
std.dev	-	0.9752	1553.8	3.54E+00	8.67E+00	5.31385	8.55E+00	8.24E+00

*Risultati algoritmo EDA e confronto*

La Tabella illustra i risultati ottenuti dall'EDA per 50 esecuzioni indipendenti dell'algoritmo, con  $\alpha = 1$  e  $\beta$  che varia tra 0,5 a 2,5 con step incrementali di 0,5. I risultati dell'EDA appaiono molto promettenti anche con un'ottimizzazione vincolata, raggiungendo ottimi sensibilmente vicini all'ottimo globale in modo affidabile. I risultati delle 50 esecuzioni sono stati confrontati con quelli degli algoritmi GA e PSO-MS, mettendo in luce scarse prestazioni su problema interessato da parte del GA. Il PSO-MS invece si comporta in maniera ottimale poiché fornisce per tutte le 50 esecuzioni indipendenti la soluzione dell'ottimo globale con una deviazione standard piuttosto ridotta. L'effetto di  $\beta$  sui risultati ottimali è sufficientemente apprezzabile. Tutti i risultati dell'EDA sono dello stesso ordine di grandezza dell'ottimo globale e sono coerenti con quelli del PSO-MS. In questo caso, un andamento sub-lineare,  $\beta = 0,5$ , sembra essere la scelta dell'iperparametro migliore per ottenere risultati ottimali con una deviazione standard leggermente bassa a parità di costo computazionale [7].

### 4.3.2 TRUSS 10

Il problema denominato “Truss 10” consiste nella ricerca del peso ottimale di una travatura a sbalzo sottoposta all’azione di forze concentrate. La travatura è composta da dieci aste con uno sbalzo complessivo di 720 in. e un’altezza di 360 in.; ad essa sono applicati due carichi nodali in direzione trasversale di valore 100 kips. Le limitazioni a cui è sottoposto il problema riguardano le tensioni agenti sulle varie travi e gli spostamenti nodali che ne derivano dall’applicazione dei carichi. Per quanto riguarda i parametri è impostata la possibilità di variare arbitrariamente con variabili continue o discontinue tutte e dieci le barre che fanno parte della struttura reticolare. Le limitazioni e le caratteristiche meccaniche sono:

- Massima sollecitazione  $\sigma$  consentita per ogni barra della travatura reticolare pari a  $\pm 25$  ksi.
- Massima deflessione nodale  $\delta$  (sia in direzione verticale che orizzontale) pari a  $\pm 2,0$  in.
- Materiale di peso unitario  $\rho$  pari a  $0,1 \text{ lb/in.}^3$  e un modulo di elasticità di 107 psi.

Il problema può essere matematicamente schematizzato così:

$$x \in \Omega$$

$$\text{Min } f(x) = \sum_{i=1}^n \rho_i l_i A_i$$

$$\text{s. t. } \sigma_i \leq \sigma_{max}$$

$$\delta_i \leq \delta_{max}$$

$$A_i^{lb} \leq A_i \leq A_i^{ub}$$

Dove  $n$  è il numero delle aste parametrizzate,  $l_i$  è la lunghezza di ogni elemento,  $\rho_i$  la densità del materiale,  $A_i^{lb}$  e  $A_i^{ub}$  sono rispettivamente i limiti inferiori e superiori del dominio delle aree preso in considerazione,  $\sigma_{max}$  e  $\delta_{max}$  rispettivamente i limiti delle tensionali e deformativi delle aste.

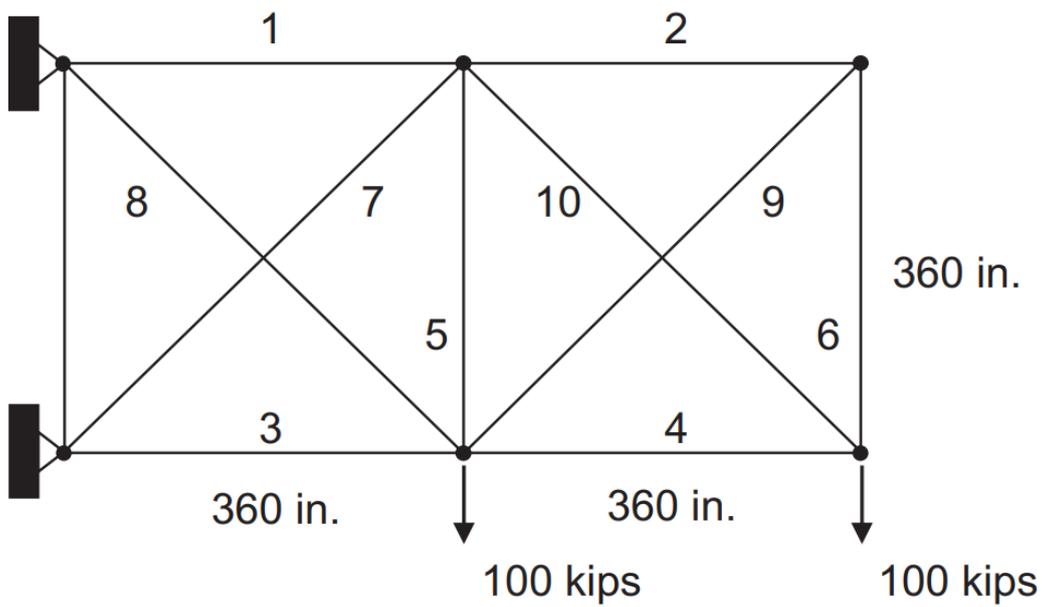


Figura 27 Geometria e schema statico del problema.

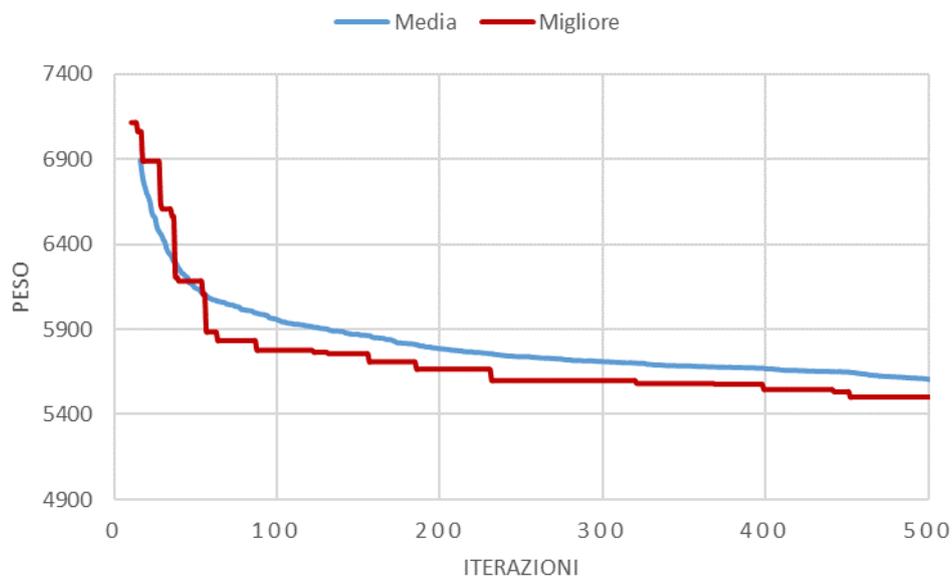
Molti studi hanno applicato varie tecniche per la risoluzione del problema di ottimizzazione: Camp et al [10]., Mahfouz [11] e Barbosa et al. [12] hanno utilizzato vari tipi di GA; Camp e Bichon [13] hanno utilizzato l'ottimizzazione delle colonie di formiche ACO; Lee e Geem [24] hanno utilizzato l'harmony search (HS), Li et al. [25] hanno utilizzato l'ottimizzazione euristica a sciame di particelle (HPSO), Sonmez [14] ha usato un algoritmo di colonia di api artificiali con un approccio di funzione di penalità adattativa (ABC-AP) e C.V. Camp e M. Farshchin [15] hanno usato l'algoritmo TLBO.

#### 4.3.2.1 Soluzione con variabili discontinue

Nel primo esempio di progettazione di una capriata a 10 barre, vengono utilizzati 41 valori discreti per le possibili aree di sezione trasversale di ogni elemento:

[1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, 33,5 in.<sup>2</sup>].

Essendo un problema su variabili discontinue per adattare l'EDA a questo tipo di problema sono state settate manualmente i valori della varianza massima  $\sigma_{j,max} = 10$  e della varianza minima  $\sigma_{j,min} = 2$ . Ciò è stato necessario per il fatto che il modello probabilistico deve generare degli indici in maniera discontinua quindi generando un numero che in seguito venga approssimato a un intero. Impostando in tale modo la varianza abbiamo imposto un limite minimo nello scostamento dall'indice di riferimento per valori discontinui.



*Andamento funzione obiettivo*

Sono state eseguite 50 analisi indipendenti con un numero massimo di 600 iterazioni. Dal grafico possiamo vedere come il punto di ottimo fosse raggiunto già con un numero inferiore a circa le 200 iterazioni. Sono stati settati due diversi valori per  $\beta$  pari a 1 e 0.5. Nella tabella sono riportati i dati estratti dalle analisi e messi a confronto con altri algoritmi di diverso genere.

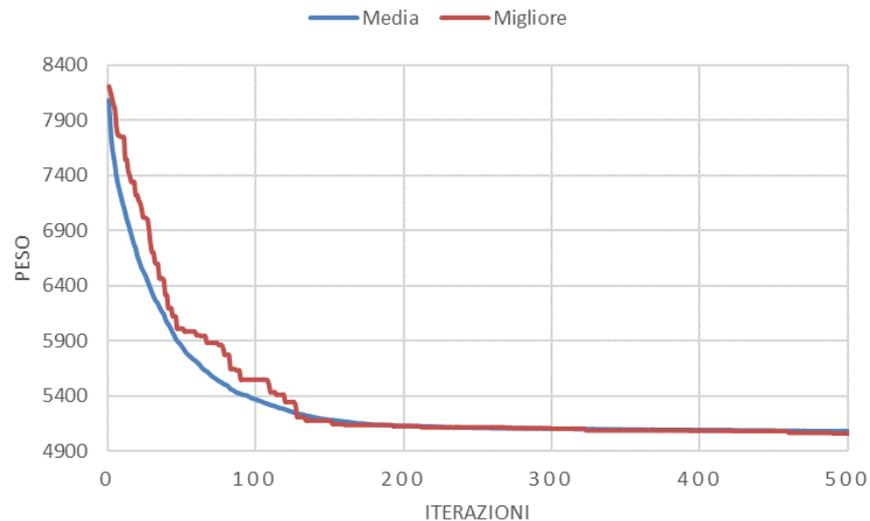
<b>Obj. Fun.</b>	<b>GA Mahfouz [11]</b>	<b>GA Barbosa et al. [12]</b>	<b>ACO Camp and Bichon [13]</b>	<b>EDA <math>\alpha=1</math> <math>\beta=0.5</math></b>	<b>EDA <math>\alpha=1</math> <math>\beta=1</math></b>
best	5490.74	5490.74	5490.74	5490.74	5491.72
mean	–	5534.98	5510.52	5565.90	5562.66
std.dev.	–	–	23.19	55.27	53.14
$N_{analyses}$	8000	200000	10000	15000	15000

*Risultati algoritmo EDA e confronto*

I risultati rivelano un buon comportamento per la risoluzione del problema anche a rispetto a un tipo di problema arduo per l'EDA essendo un algoritmo performante su variabili continue.

## Soluzione con variabili continue

Il secondo metodo di risoluzione del problema “Truss 10” viene impostato permettendo che le aree della sezione trasversale possano variare da 0,1 in.<sup>2</sup> a 35,0 in.<sup>2</sup>.



### *Andamento funzione obiettivo*

Sono state eseguite 50 analisi indipendenti con un numero massimo di 500 iterazioni. Dal grafico possiamo vedere come il punto di ottimo fosse raggiunto già con un numero inferiore a circa le 200 iterazioni. Sono stati settati due diversi valori per  $\beta$  pari a 1 e 0.5. Nella tabella sono riportati i dati estratti dalle analisi e messi a confronto con altri algoritmi di diverso genere.

<b>Obj. Fun.</b>	<b>GA Camp et al. [10]</b>	<b>ABC-AP Sonmez [16]</b>	<b>TLBO [15]</b>	<b>EDA <math>\alpha=1</math> <math>\beta=0.5</math></b>	<b>EDA <math>\alpha=1</math> <math>\beta=1</math></b>
best	5076.31	5060.88	5060.97	5065.02	5064.18
mean	-	-	5064.81	5077.98	5082.92
std.dev.	-	-	6.37	9.29	11.07
$N_{analyses}$	15000	500000	13767	12500	12500

### *Risultati algoritmo EDA e confronto*

I risultati elaborati dall’EDA sono molto promettenti poiché, pur non raggiungendo un punto di ottimo migliore degli altri algoritmi, vediamo che con un basso numero di iterazioni ci avviciniamo con buona approssimazione al punto di ottimo.

## 5 CASO STUDIO: STRUTTURA RETICOLARE IN ACCIAIO

### 5.1 INTRODUZIONE

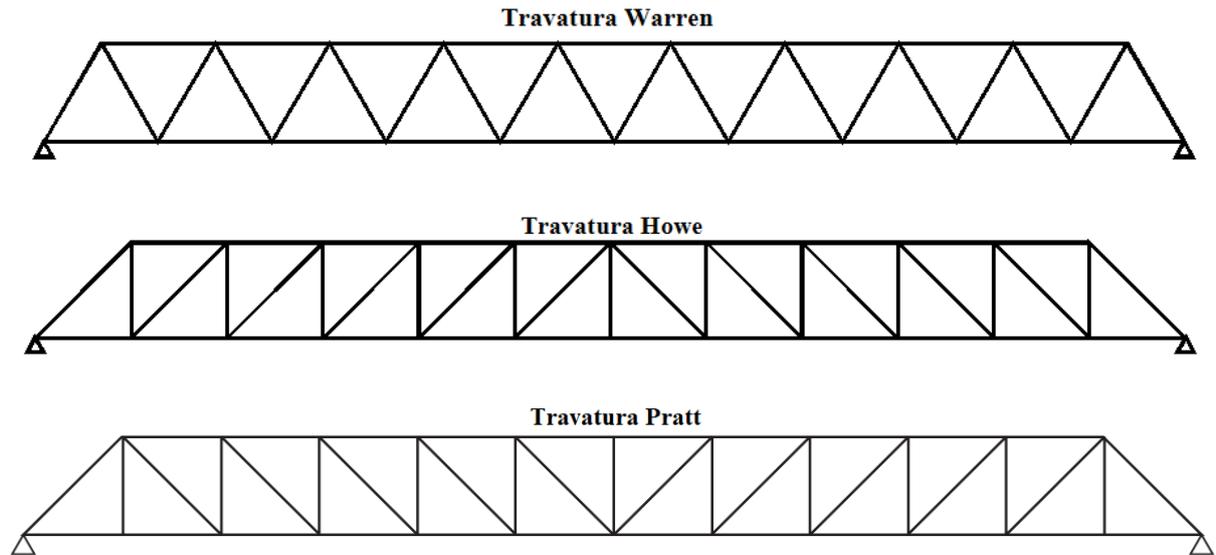
«Una struttura reticolare è un particolare sistema strutturale in cui tutte le travi (aste) sono collegate tra loro alle estremità da cerniere interne (nodi) e sono vincolate a terra tramite cerniere o carrelli esterni applicati ai nodi. Se i vincoli (esterni ed interni) sono perfetti, cioè senza attrito, e le forze esterne sono applicate solo nei nodi, allora le aste risultano soggette solo a sforzi assiali di trazione o compressione.» Casini P., Vasta M., Scienza delle costruzioni, 2016, Città Studi Edizioni

Una struttura reticolare è una struttura rigida e leggera costruita grazie all'incastro di montanti seguendo ben definiti schemi geometrici, la quale permette di coprire ampie luci, con una soluzione affidabile sia dal punto di vista strutturale che di peso, quindi di anche di costi. Si viene così a creare il sistema reticolare spaziale denominato "Space Frame", che si rivela per molte ragioni il più adeguato, fra i vari schemi strutturali, per la progettazione di strutture portanti di copertura in quanto consente anche una produzione altamente. [17]

Il materiale di uso più comune è l'acciaio, per le sue caratteristiche di resistenza, di duttilità, di leggerezza e di semplicità di giunzione mediante bullonature o saldature.

Le travature reticolare sono strutture composte da un insieme di aste complanari, vincolate ai nodi in modo da costituire, globalmente, un elemento con un'alta resistenza e bassa deformabilità. Essa è formata da due elementi continui chiamati correnti, e da un'anima scomposta in elementi lineari, i quali posso essere inclinati a seconda del comportamento che vogliamo conferire alla reticolare. In genere gli elementi verticali vengono denominati montanti mentre, quelli inclinati, vengono denominati diagonali; il comportamento di questi elementi è eguale a quello di un elemento sottoposto a soli sforzi di trazione e compressione. [17]

Esistono diverse tipologie di travature reticolari, differenti tra di loro per geometria:



*Figura 28 Tipologie di travature reticolari.*

L'utilizzo di questo sistema strutturale ha trovato largo impiego per la realizzazione di coperture di grandi luci per strutture ad uso industriale e sportivo. Le necessità hanno portato a un crescente bisogno di spazi liberi per lo svolgersi delle attività lavorative o lo stoccaggio di manufatti di grandi dimensioni. Ciò ha portato all'impiego di travature reticolari nella realizzazione di coperture che possono arrivare oltre i 30 metri. Tali strutture con la loro leggerezza e l'alta resistenza permettono di coprire luci di tali dimensioni offrendo a loro volta una soluzione economica e sicura e che rispetti i vincoli architettonici imposti.



*Figura 29 Copertura dell'acciaieria ILVA a Taranto, luce complessiva 254 metri.*

## 5.2 DESCRIZIONE CASO STUDIO

La struttura presa in oggetto è la copertura industriale di caso studio generico con una luce fissa di 300 metri e una lunghezza di 800 metri. Le colonne hanno un'altezza fissa di 40 metri con una spaziatura modulare fissa. La copertura ha una dimensione complessiva di 24000 m<sup>2</sup>.

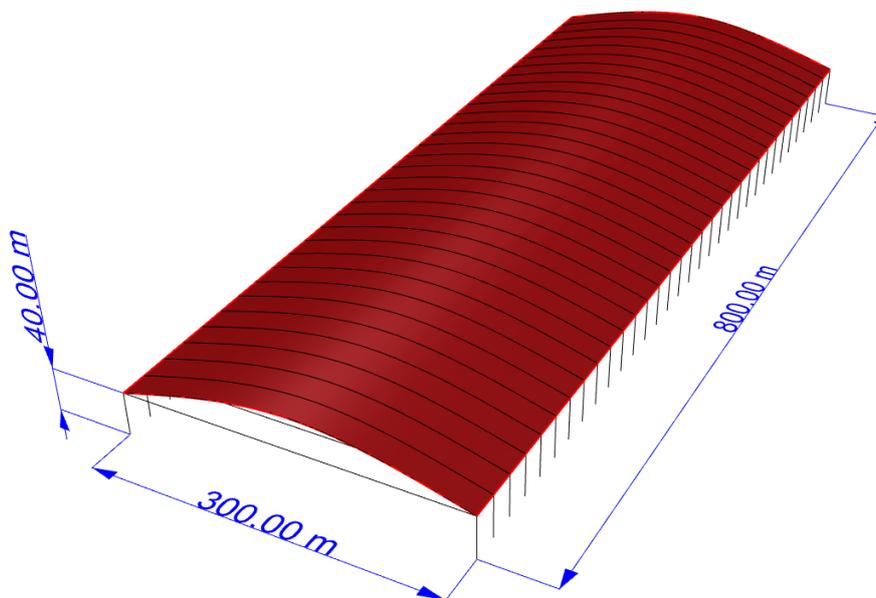


Figura 30 Caso studio: Dimensioni struttura

La struttura è costituita da un sistema di controventatura verticale a croci di Sant'Andreas nella direzione longitudinale maggiore e da un sistema di controventatura di falda al fine di garantire un comportamento a piano rigido della copertura. La trave reticolare è costituita da un insieme di montanti e correnti a inclinazione variabile. Sono inoltre presenti travature reticolari longitudinali al fine di irrigidire la copertura e permettere la ripartizione degli sforzi. Il processo di ottimizzazione è stato attuato dividendo la struttura in moduli delle dimensioni di longitudinale di 100 metri.

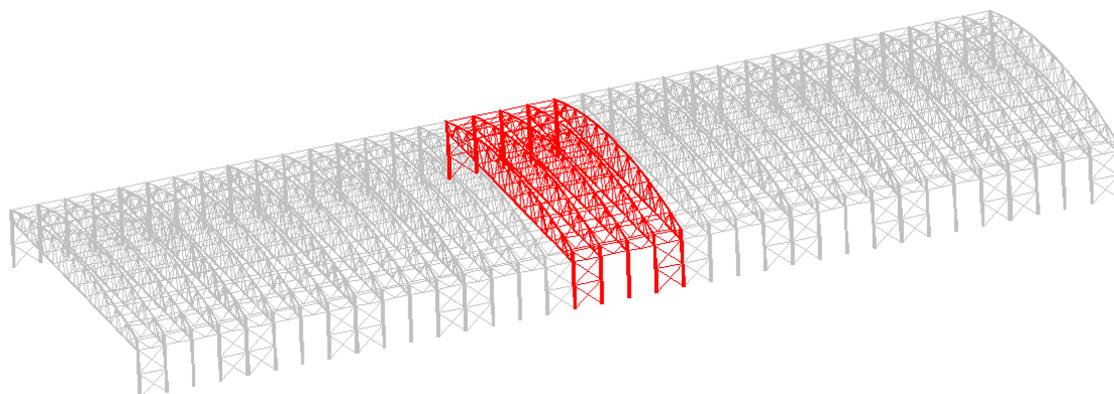
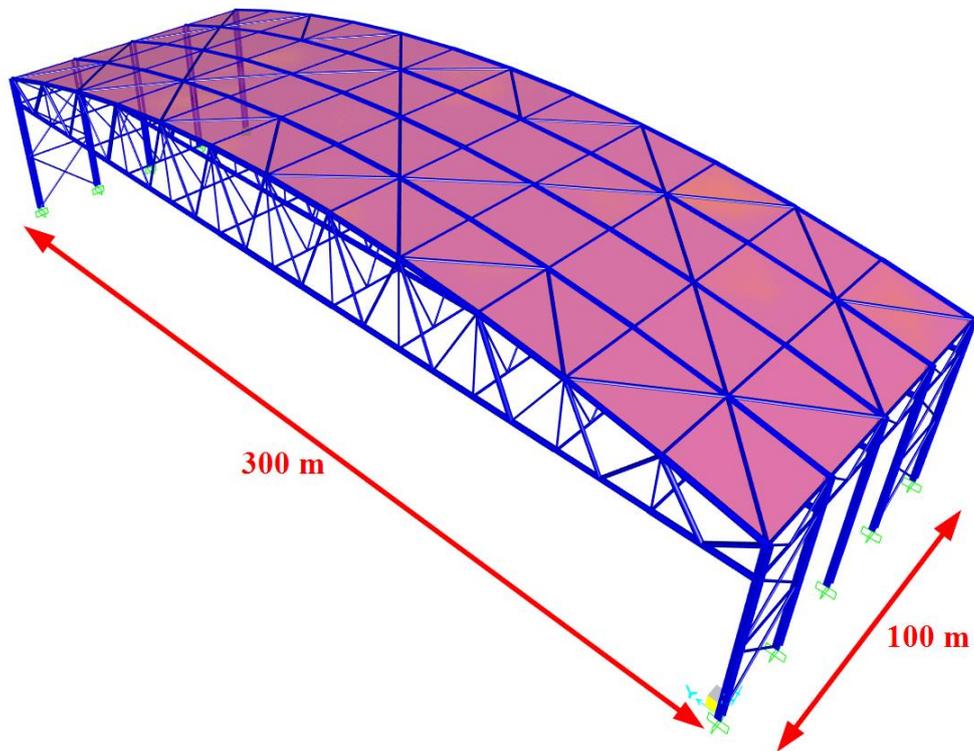


Figura 31 Schematizzazione della struttura

Il modulo preso in considerazione riesce a cogliere il comportamento strutturale globale della struttura per le azioni statiche presenti su di essa. Sono state prese quindi in considerazione 4 campate con colonne controventate agli estremi per assorbire i carichi orizzontali.



*Figura 32 Dimensioni caso studio*

Sono state condotte due tipologie di analisi la prima fissando la geometria della copertura e impostando come parametri variabili le sezioni che la compongono. Le sezioni sono state selezionate da una lista di tubolari CHS di diametro e spessore variabile. Nel secondo tipo di analisi sono stati impostati come parametri dell'ottimizzazione diversi insiemi di parametri geometrici e meccanici della struttura.

### 5.2.1 SCHEMA STATICO

La struttura è stata vincolata alla base di ogni colonna con un incastro, bloccando la traslazione e la rotazione nelle tre direzioni X, Y e Z.

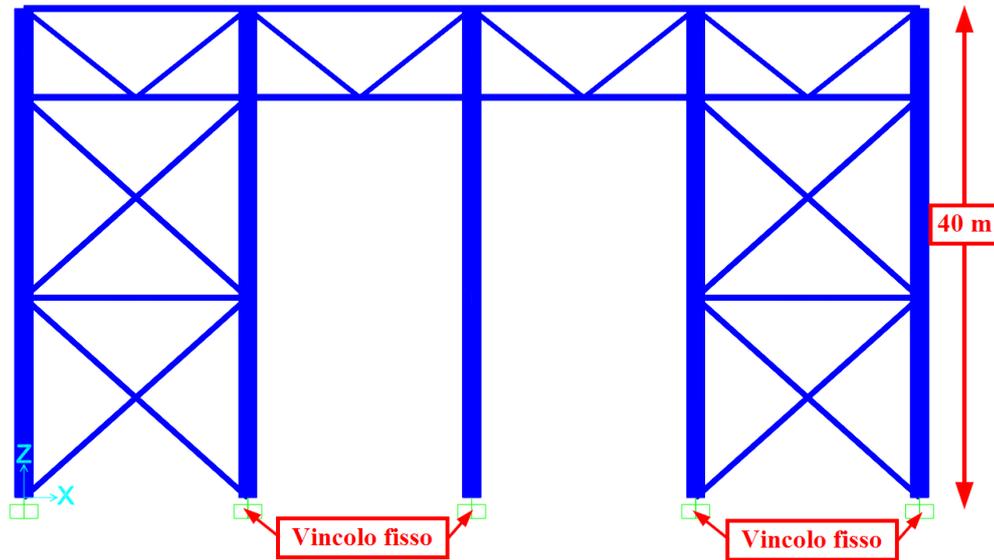


Figura 33 Vincoli della struttura.

Per lo schema statico dei vincoli interni sono state considerate connessioni saldate tra le varie sezioni tubolari per cui sono stati considerati degli incastrati tra i vari elementi strutturali. Infine per modellare la copertura sono stati considerati degli elementi "Load patch" che distribuiscono i carichi della copertura sulla travatura reticolare.

### 5.2.2 MATERIALE

È stato considerato, per l'analisi della struttura, un acciaio strutturale S355 con le seguenti caratteristiche meccaniche:

- Densità  $\delta = 76,98 \text{ kN/m}^3$
- Modulo di elasticità  $E = 210000 \text{ N/mm}^2$
- Modulo di Poisson  $\nu = 0,3$
- Resistenza a snervamento  $f_{yk} = 355 \text{ N/mm}^2$
- Resistenza a ultima  $f_{uk} = 510 \text{ N/mm}^2$

### 5.2.3 CARICHI

Per simulare il carico della copertura è stato applicato un carico variabile distribuito **L** sulla copertura di valore pari a  $-1,47 \text{ kN/m}^2$  in direzione Z. Per il peso proprio della struttura **D** è stata applicata un valore di densità  $\delta$  dell'acciaio pari a  $76,98 \text{ kN/m}^3$ .

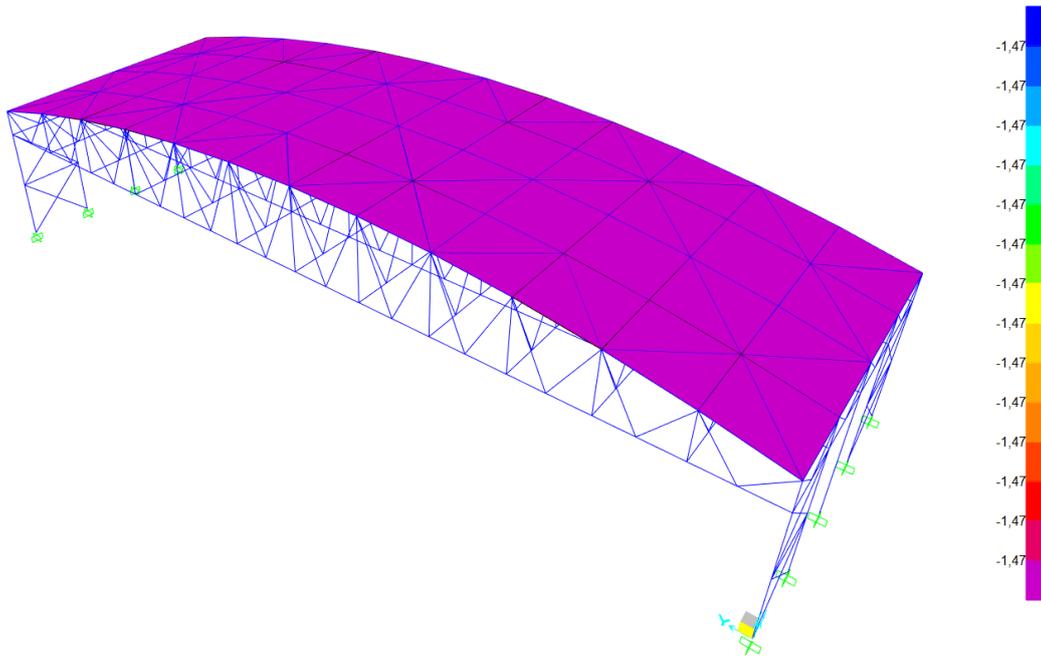


Figura 34 Carico variabile applicato alla copertura  $\text{kN/m}^2$

### 5.2.4 COMBINAZIONI

I carichi sono stati combinati secondo la normativa americana ASCE 7-05. Le combinazioni prese in considerazioni sono le seguenti:

#### ULS

1.  $1.4\mathbf{D}$
2.  $1.2\mathbf{D}+1.6\mathbf{L}$

#### SLS

1.  $1.0\mathbf{D}$
2.  $1.0\mathbf{D}+1.0\mathbf{L}$

Dove **D** è il peso proprio della struttura e **L** è il carico variabile applicato sulla copertura.

### **5.2.5 VERIFICHE**

Gli elementi strutturali sono stati verificati seguendo la norma americana AISC 360-16. Per la verifica degli elementi è stato considerato quindi il metodo del “Effective length” che prevede di considerare la lunghezza di inflessione effettiva degli elementi. Tali verifiche sono state impostate come vincolo per l’ottimizzazione della struttura.

### 5.3 MODELLO PARAMETRICO

Il problema strutturale è stato elaborato grazie all'interazione di un insieme di software specialistici che hanno permesso la risoluzione del problema. In primo luogo, è stato possibile gestire la geometria della struttura grazie all'ambiente di progettazione 3D chiamato Rhinoceros mentre le analisi strutturali sono state eseguite grazie al programma agli elementi finiti Sap2000. La gestione del flusso di dati tra i due programmi è stata possibile grazie al software Grasshopper, un programma che permette il "visual coding" all'interno di Rhinoceros. All'interno di Grasshopper è stato possibile parametrizzare ad interim la geometria della struttura, creare un modello FEM, elaborare i risultati delle analisi strutturali e gestire il processo di ottimizzazione. L'interazione di questi potenti software ha dato il grosso vantaggio di gestire e controllare costantemente il processo di ottimizzazione, oltre alla possibilità di parametrizzare in ogni suo aspetto la struttura e di elaborare un unico modello per la risoluzione del problema di ottimizzazione. Nei seguenti paragrafi sarà spiegato il funzionamento del modello di ottimizzazione e l'interazione tra i vari programmi.

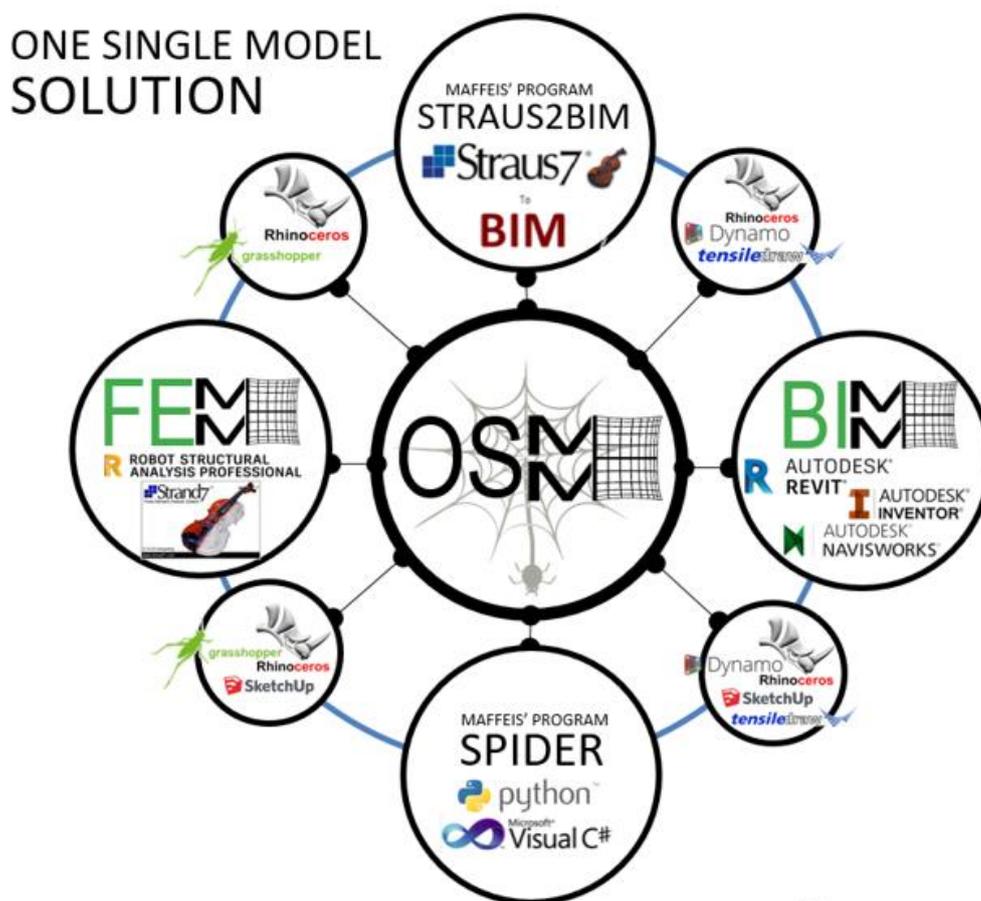
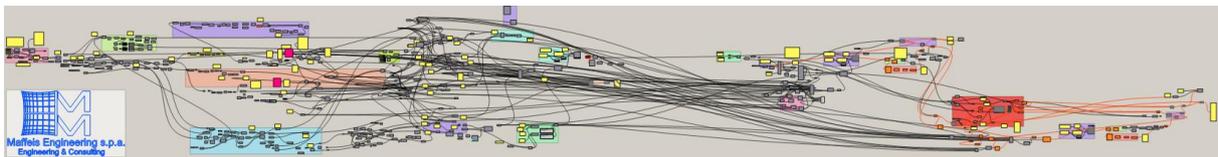


Figura 35 FeMM Maffeis interoperabilità

### 5.3.1 PROCESSO DI OTTIMIZZAZIONE

Il processo di ottimizzazione è stato sviluppato all'interno del programma Grasshopper il quale ha permesso di creare un unico modello per la risoluzione del problema e di impostare come parametri dell'ottimizzazione sia le caratteristiche geometriche della struttura che le caratteristiche meccaniche, in particolare i profili delle sezioni. La creazione del modello è stata possibile grazie alle varie componenti che hanno permesso lo svolgersi delle seguenti operazioni:

1. Generazione dei parametri di input tramite Latin Hypercube Sampling.
2. Creazione del modello geometrico in Rhino.
3. Creazione del modello FEM e gestione dei risultati delle analisi grazie a componenti di Grasshopper sviluppate da Maffeis SpA.
4. Sviluppo del processo di ottimizzazione tramite EDA grazie a una componente di Grasshopper che permette la lettura del codice scritto in linguaggio Python.
5. Sviluppo di un processo iterativo grazie alla componente Anemone disponibile sulla piattaforma Food4Rhino.



*Figura 36 Ambiente Grasshopper.*

Il processo di ottimizzazione è stato sviluppato gestendo i risultati dell'analisi agli elementi finiti, permettendo quindi di impostare come funzione obiettivo il peso proprio della struttura e come vincoli strutturali le verifiche condotte secondo la normativa AISC 360-16.

Il problema è così definito:

$$x \in \Omega$$

$$\text{Min } f(x) = \sum_{i=1}^n \rho_i l_i A_i$$

$$\text{s. t. } \frac{\sigma_i}{\sigma_{max}} - 1 \leq 0$$

$$\frac{\delta_i}{\delta_{max}} - 1 \leq 0$$

$$A_i^{lb} \leq A_i \leq A_i^{ub}$$

Dove  $n$  è il numero totale di aste della struttura,  $l_i$  è la lunghezza di ogni elemento,  $\rho_i$  la densità del materiale,  $A_i^{lb}$  e  $A_i^{ub}$  sono rispettivamente i limiti inferiori e superiori del dominio delle aree preso in considerazione da una lista di profili predefinita,  $\frac{\sigma_i}{\sigma_{max}}$  e  $\frac{\delta_i}{\delta_{max}}$  sono rispettivamente i limiti delle tensionali e deformativi generati come dati in uscita dai risultati dell'analisi agli elementi finiti.

La prima fase del processo di ottimizzazione prevede la generazione e la valutazione degli individui appartenenti alla prima popolazione da valutare. Consideriamo di generare un numero pari di  $N$  individui tramite Latin Hypercube Sampling, la prima fase così si compone:

Per  $N$  volte:

- Creazione del modello geometrico.
- Creazione del modello FEM.
- Risoluzione del modello FEM.
- Gestione dei risultati delle analisi condotte.
- Incameramento dei risultati.

Il processo è quindi condotto per ogni individuo così da valutarne le caratteristiche iniziali e inviare i dati iniziali all'ottimizzatore. La seconda fase prevede il vero e proprio processo di ottimizzazione. Considerando di iterare il processo di ottimizzazione per  $K$  volte e assumendo

una selezione del 50% degli individui per troncamento, il processo si compone dalle seguenti fasi.

Per K volte:

- Valutazione della popolazione in entrata e generazione di una nuova popolazione composta da  $\frac{N}{2}$  individui.

Per  $\frac{N}{2}$  volte:

- Creazione del modello geometrico.
- Creazione del modello FEM.
- Risoluzione del modello FEM.
- Gestione dei risultati delle analisi condotte.
- Incameramento dei risultati.

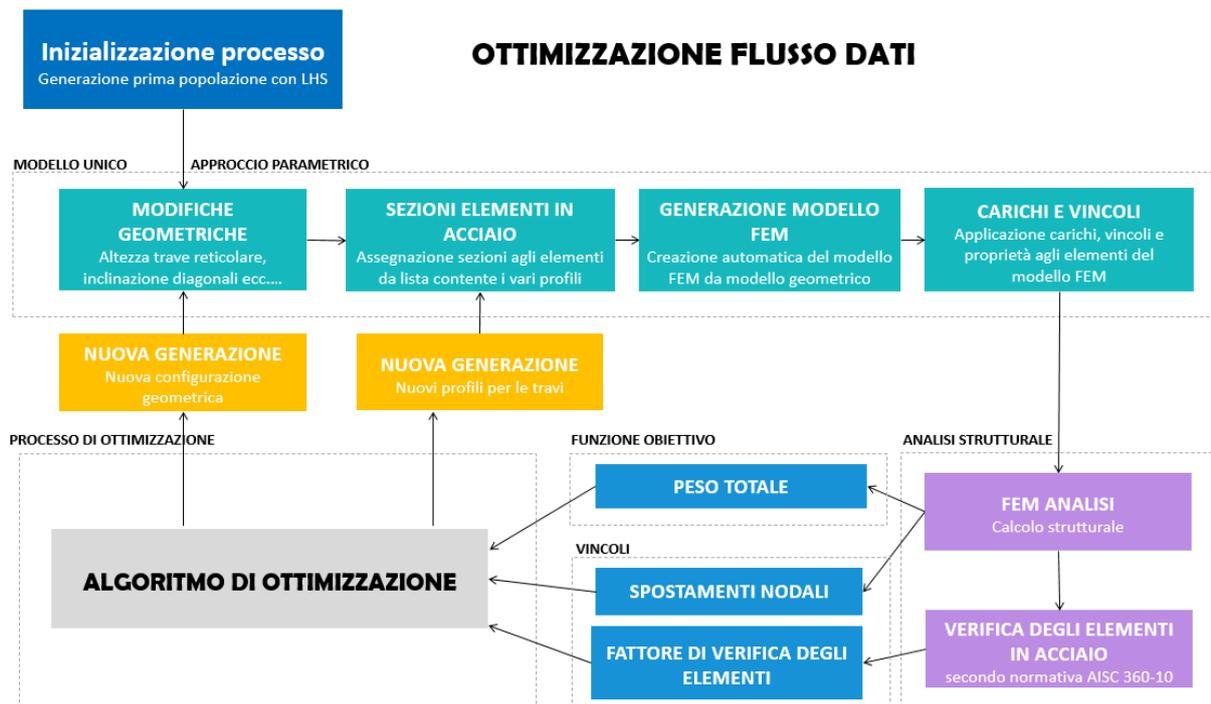
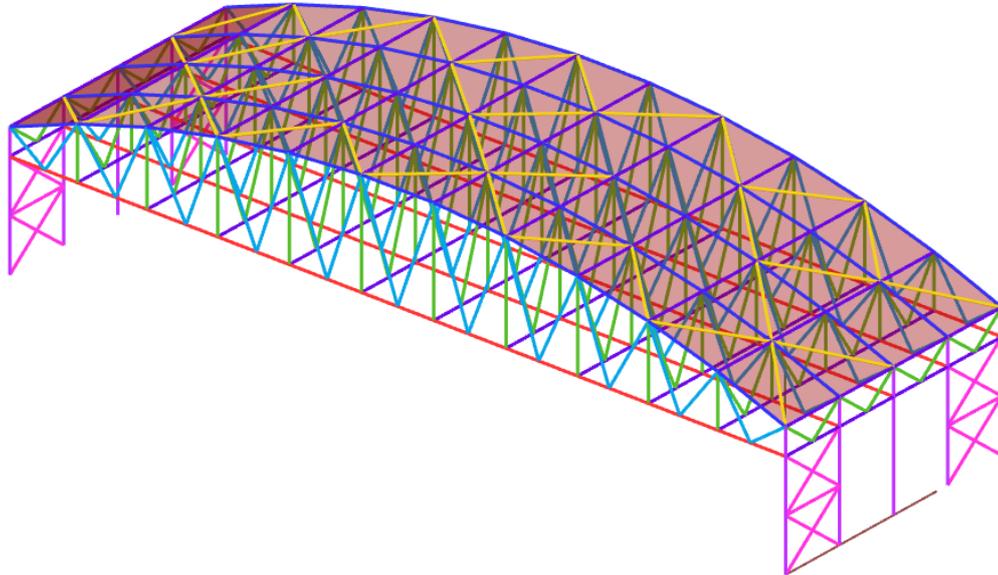


Figura 37 Flusso dei dati per il processo di ottimizzazione e le varie operazioni.

### 5.3.2 PARAMETRIZZAZIONE GEOMETRICA

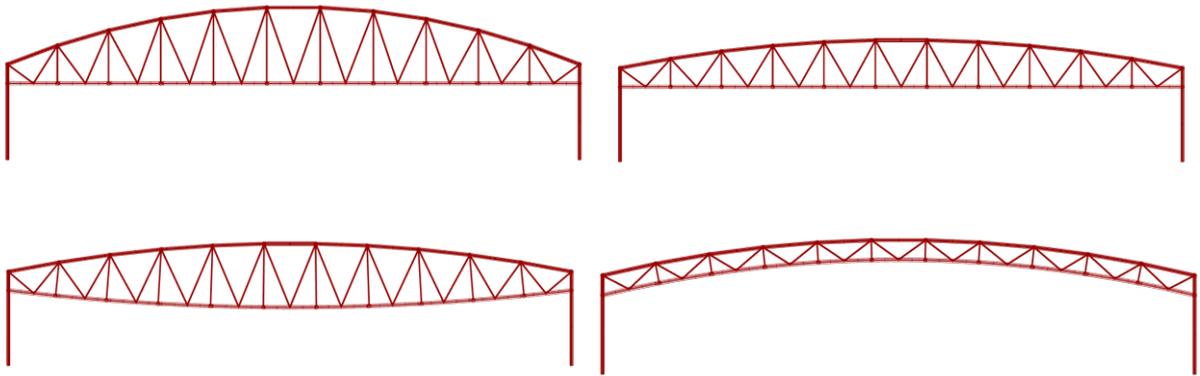


*Figura 38 Modello geometrico*

Il processo di creazione e parametrizzazione del modello geometrico è stato realizzato all'interno di Grasshopper grazie alle sue componenti che permettono un'interazione con il software Rhinoceros. La parametrizzazione è stata effettuata introducendo le caratteristiche geometriche fissate:

- Altezza delle colonne 40m.
- Lunghezza della campata della trave reticolare di copertura 300m.
- Larghezza del modulo geometrico 100m.

Partendo da questi dati fondamentali è stato possibile generare un modello parametrizzato in ogni sua caratteristica come l'altezza e le dimensioni della travatura reticolare e l'interasse dei diagonal. Ogni elemento geometrico è stato affidato a un gruppo a seconda della sua funzione in modo tale da gestire in maniera ottimale l'affidamento delle caratteristiche meccaniche. Inoltre, sono stati utilizzati degli elementi mesh per ricreare la copertura della struttura.



*Figura 39* Possibili configurazioni geometriche: variazione della curvatura del corrente superiore e inferiore.



*Figura 40* Possibili configurazioni geometriche: variazione inclinazione dei diagonali.

### 5.3.3 PARAMETRIZZAZIONE MODELLO FEM

Nello ambiente Grasshopper è stato creato, grazie ai tools sviluppati dalla Maffeis SpA, un processo di modellazione per la creazione e l'analisi del modello agli elementi finiti. Per mezzo di un pacchetto interno all'azienda denominato "FeMM" è stato possibile gestire il processo di affidamento delle caratteristiche meccaniche alle varie aste, l'assegnazione dei vincoli, la scelta del tipo di analisi e le combinazioni di carico.

Il modello FEM è stato impostato fissando dei parametri variabili per la struttura e dei parametri fissi quindi il processo ha attraversato una fase iniziale di creazione del modello e una fase secondaria di costante aggiornamento e analisi. Sono stati scelti come parametri fissi i seguenti:

- Materiale acciaio strutturale S355.
- Vincoli fissi alla base delle colonne.
- Vincoli interni incastri tra le varie aste.
- Load Patch mono direzionali tra le campate della struttura.
- Combinazioni di carico secondo ASCE 7-10.
- Carico variabile **L** imposto alla struttura pari a  $-1,47 \text{ kN/m}^2$  in direzione Z.

Sono stati impostati come parametri variabili nella generazione del modello FEM l'assegnazione dei profili alle varie aste. Per far ciò è stato assegnato ad ogni asta un gruppo a seconda delle sue funzionalità nella struttura (diagonali, montanti, correnti ecc...). Dopo di che, è stato possibile assegnare una sezione all'asta scegliendola da una lista di profili preimpostata e inseguito aggiornare iterativamente il processo. Il processo di selezione del profilo avviene grazie a un'indicizzazione della lista e quindi alla scelta del profilo in base al proprio indice. In tale modo diventa di fondamentale importanza la disposizione dei profili all'interno della lista poiché queste determinano la rigidezza e il peso della struttura. Nel caso specifico analizzato è stato preferibile disporre le sezioni, dalla più piccola alla più grande, a seconda delle dimensioni areali del profilo questo perché tale caratteristica è sia dominante nella variazione della funzione obiettivo che nella valutazione dei vincoli.

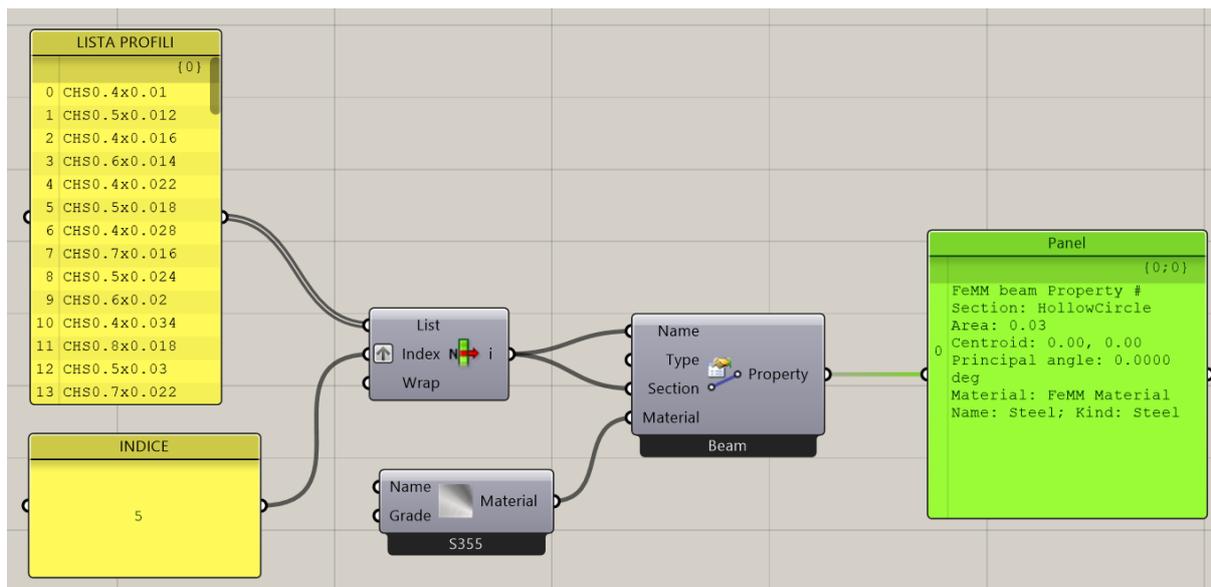


Figura 41 Assegnazione delle proprietà meccaniche.

Dopo aver assegnato le varie caratteristiche al modello è stato possibile creare o aggiornarlo, analizzarlo e verificarlo esportandolo da Rhinoceros al programma di analisi agli elementi finiti Sap2000. Tutto ciò è stato possibile grazie ai tool di Maffeis che hanno permesso anche di creare un processo iterativo dal momento che, ogni volta che i parametri del modello vengono modificati, in automatico il modello e l'analisi viene aggiornata.

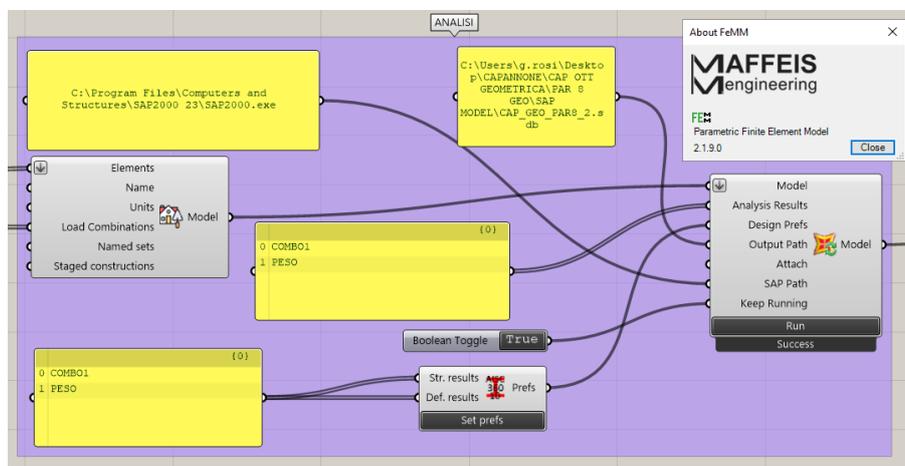


Figura 42 Analisi strutturale con tool FeMM.

Una volta eseguita l'analisi e verificata la struttura i risultati vengono caricati nell'ambiente Grasshopper dove è possibile gestirli e impostare la valutazione dei vincoli. Per il calcolo della funzione obiettivo è stato possibile leggere le reazioni alla base di ogni singola colonna per il carico "Dead" che rappresenta il peso proprio della struttura, da queste è stata eseguita una sommatoria per le reazioni in direzione Z così da calcolare il peso complessivo della struttura. Per quanto riguarda i vincoli strutturali, Sap2000 restituisce come valore in uscita dal

verificatore interno il massimo rapporto tra la sollecitazione agente  $S_j$  e l'azione resistente  $R_j$  tra le varie verifiche impostate secondo normativa per ogni singolo elemento. Sono stati impostati i valori della normativa AISC 360-10 per la valutazione delle verifiche strutturali da condurre su ogni singolo elemento.

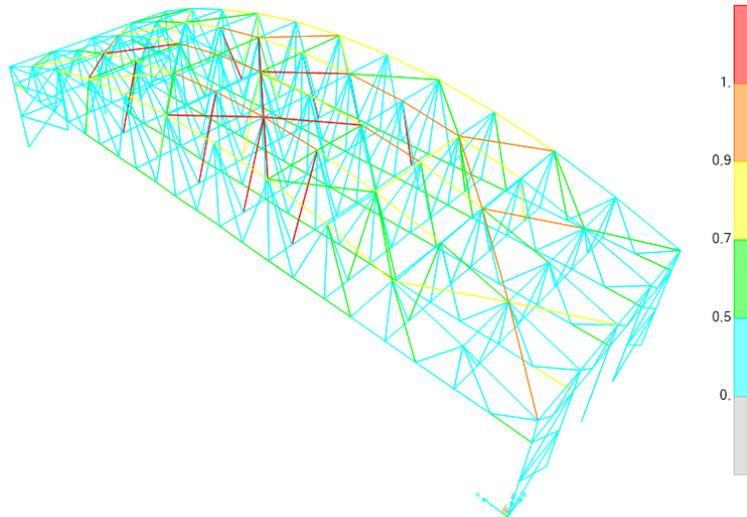


Figura 43 Risultati verifiche modello FEM in Sap2000

AISC360-10 steel prefs			
Framing type	SMF	Consider deflection	<input type="checkbox"/>
Seismic design category	D	DL deflection limit (Lf)	120
Design provision	LRFD	SDL+LL deflection limit (Lf)	120
Notional load coeff.	0.002	LL deflection limit (Lf)	360
Phi bending	0.90	Total deflection limit (Lf)	240
Phi compression	0.90	Total camber limit (Lf)	240
Phi tension yielding	0.90	Pattern live load factor	0.75
Phi tension fracture	0.75	Demand/Capacity	0.95
Phi shear	0.90	Multi-response case design	Envelopes
Phi shear sort web. rolled I	1.00	Analysis method	Effective length
Phi torsion	0.90	Second order method	General 2nd order
Ignore seismic code	<input type="checkbox"/>	Stiffness reduction method	Tau b fixed
Ignore special seismic load	<input type="checkbox"/>	Design system rho	1.00
Doubler plate is plug welded	<input checked="" type="checkbox"/>	Design system sds	0.50
HSS welding type	ERW	Design system R	8.00
Reduce HSS thickness	<input type="checkbox"/>	Design system Omega0	3.00
Auto generate check combs	<input checked="" type="checkbox"/>	Design system Cd	5.50
Importance factor	1.00		

Defaults    Cancel    Ok

Figura 44 Impostazione verifiche normativa AISC 360-10.

Per semplicità e per diminuire l'onere computazionale avendo un elevato numero di risultati è stato ridotto il problema strutturale complessivo ad un'unica valutazione dei vincoli. Considerando di avere un numero di aste J la valutazione dei vincoli per le aste è così calcolata:

vincolo asta  $j$  – esima 
$$\frac{S_j}{R_j} - 1 \leq 0$$

A seguito della valutazione delle aste avremo un numero  $K$  di aste verificate e un numero  $H$  di aste che non rispettano i limiti delle verifiche. Da questa valutazione è possibile ricavare la valutazione globale della struttura nel seguente modo:

*vincolo della struttura*

$$n_h \cdot \sum_{h=1}^H \left( \frac{S_h}{R_h} - 1 \right) \leq 0$$

Dove  $S_j$  e  $R_j$  sono rispettivamente l'azione sollecitante e l'azione resistente dell'asta  $h$ -esima mentre  $n_h$  è il numero delle aste che non rispettano le verifiche strutturali. Grazie a questo espediente è stato possibile ridurre sensibilmente l'onere computazionale dell'ottimizzatore gestendo così un unico vincolo e riducendo la molle di flusso dati all'interno del processo iterativo.

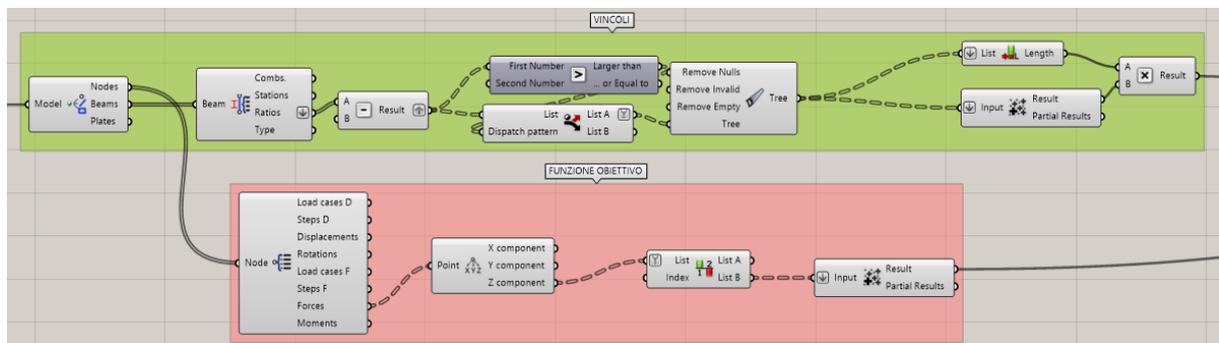


Figura 45 Script per calcolo della funzione obiettivo e analisi dei vincoli strutturali.

### 5.3.4 OTTIMIZZATORE

Per attuare il processo di ottimizzazione sono state utilizzate tre tools fondamentali per rendere realizzabile il processo. È stato di grande importanza trovare una componente che permettesse l'esecuzione del codice scritto in linguaggio Python. La componente fornita da Grasshopper per la scrittura di codici Python si avvale dell'implementazione di linguaggio IronPython che funziona sulle implementazioni del .NET Framework (C#). Il problema sorge al momento dell'import delle librerie Numpy [18] che si basano su linguaggio nativo C quindi incompatibili con IronPython [19]. Per permettere l'esecuzione del codice è stato quindi necessario utilizzare la componente GH\_CPython [20], che legandosi al software Anaconda [21], interpreta il linguaggio Python ed esegue il processo di ottimizzazione.

Il processo è stato implementato utilizzando due componenti scritte appositamente per l'ottimizzazione tramite GH\_CPython e una componente chiamata Anemone [22] per l'esecuzione dei vari loop.

La prima componente scritta in GH\_CPython è stata una tools che permettesse l'esecuzione del Latin Hypercube Sampling. Per fare ciò è stata importata la libreria Scipy [23] ed eseguito il codice per il Latin Hypercube Sampling impostando come parametri in entrata il dominio per ogni singolo parametro e il numero di individui da generare per creare la popolazione iniziale. I dati in entrata vengono letti dalla componente come matrici e in uscita avremo la popolazione iniziale da analizzare.

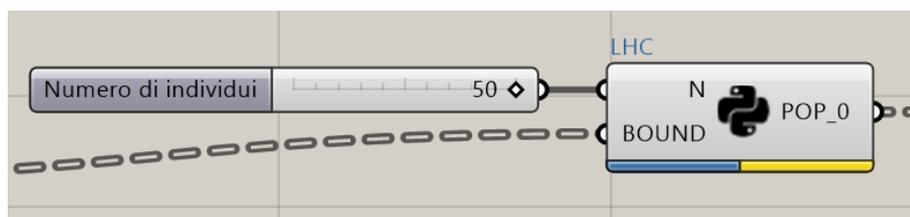


Figura 46 Componente Grasshopper per Latin Hypercube Sampling. In entrata il numero di individui da generare e la matrice contenente il dominio per ogni singolo parametro, in uscita la popolazione iniziale.

La seconda componente realizzata è l'ottimizzatore basatosi sull'algoritmo EDA. La componente per svolgere il processo di ottimizzazione ha bisogno dei seguenti dati in entrata:

- Numero di individui della popolazione (Population\_Size).
- Numero di parametri (Dimensions).
- Numero di iterazioni massime impostato (Iteration\_Number).
- Valore dell'iterazione in esecuzione (Iteration\_Count).

- Matrice contenete i parametri della popolazione da valutare (Population).
- Vettore contenete i valori della funzione obiettivo (Objective\_Function).
- Valutazione dei vincoli del problema (Constraints).
- Matrice contenete il dominio per ogni singolo parametro (Bound).

La componente restituisce i seguenti dati in uscita:

- Matrice contenete i parametri della nuova popolazione (New\_Pop).
- Matrice contenete i parametri della popolazione selezionata per la generazione del nuovo modello probabilistico (Population\_Valutated).
- Vettore contenete i valori della funzione obiettivo della popolazione selezionata (Objective\_Function\_Old\_Pop).
- Valutazione dei vincoli della popolazione selezionata (Constraints\_Old\_Pop).
- Matrice contenente i parametri degli individui “feasible” (Feas\_Results).

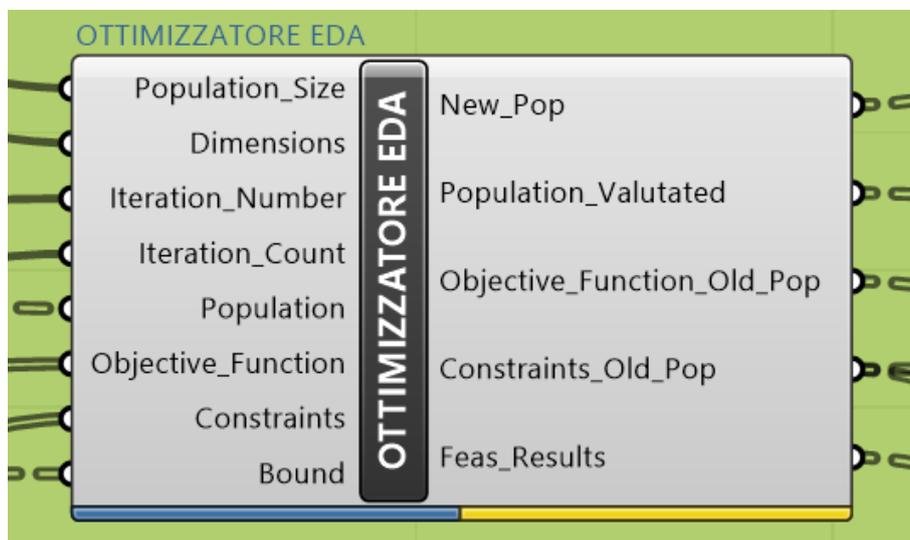


Figura 47 Componente Grasshopper per ottimizzazione tramite algoritmo EDA.

Per un corretto funzionamento dell'ottimizzatore è richiesto che i dati in uscita della popolazione selezionata, con le relative valutazioni dei vincoli e della funzione obiettivo,

rientrano all' interno di esso per rivalutare tutti gli individui in funzione della funzione di penalizzazione e iterare il processo di generazione dei nuovi individui.

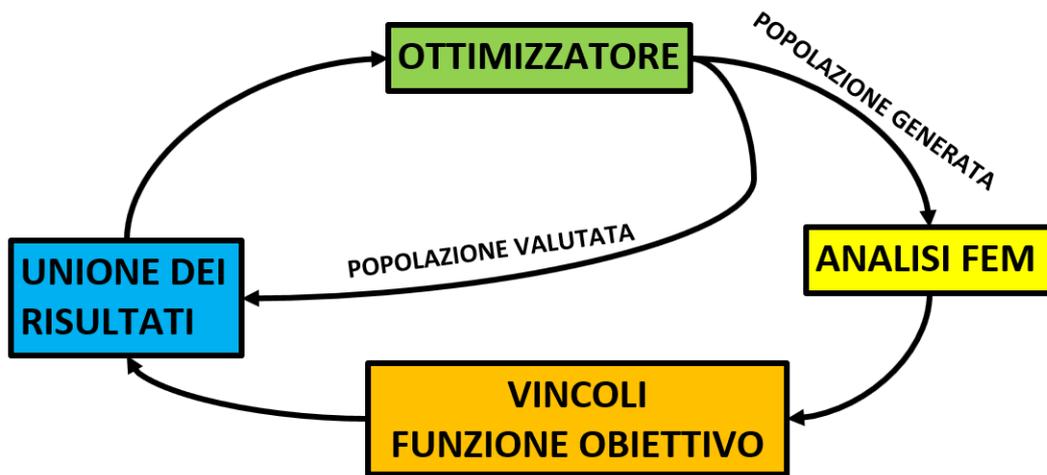


Figura 48 Flusso di dati dell'ottimizzatore.

Infine, per poter creare un processo di iterazione del flusso dati è stata utilizzata la componente per la generazione loop all'interno dell'ambiente Grasshopper chiamata Anemone [22]. La componente ha permesso di impostare il numero massimo di iterazioni da effettuare, l'introduzione della prima popolazione di individui generata tramite Latin Hypercube Sampling, di registrare i dati di passaggio a ogni iterazione e di contare il numero di iterazioni in atto.

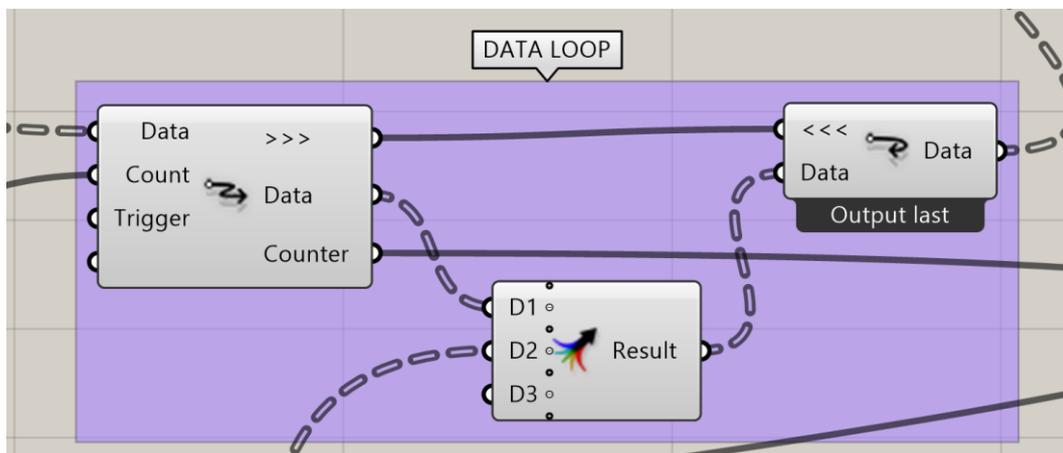
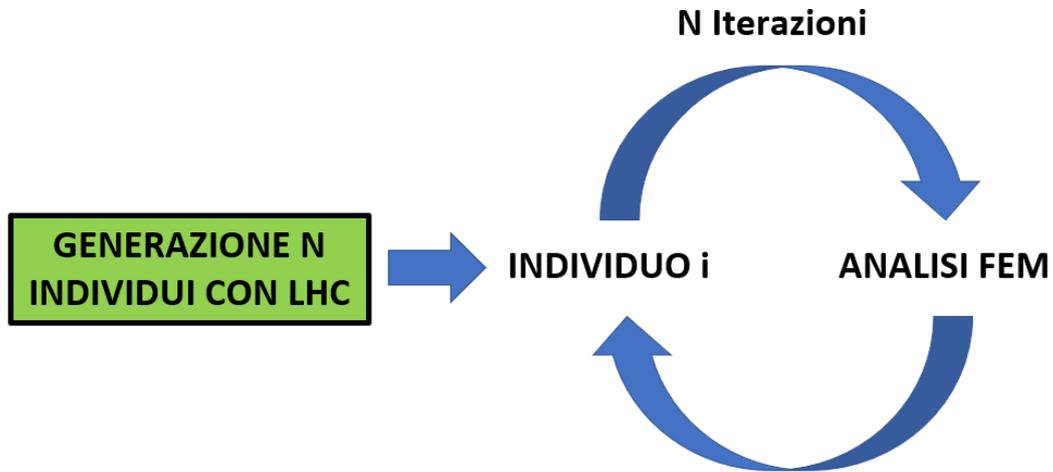


Figura 49 Componente Anemone che permette di creare un processo iterativo.

La componente è stata utilizzata per fare una sequenza di loop in modo tale da condurre le analisi per ogni singolo individuo nella fase iniziale di valutazione della popolazione iniziale incamerando i dati da inviare all'ottimizzatore.



*Figura 50 Processo iniziale di valutazione di N individui generati tramite Latin Hypercube Sampling.*

Dopo di che, il processo di ottimizzazione prevede l'inizio del processo iterativo di ottimizzazione, il quale al suo interno contiene un loop per l'analisi degli individui generati.

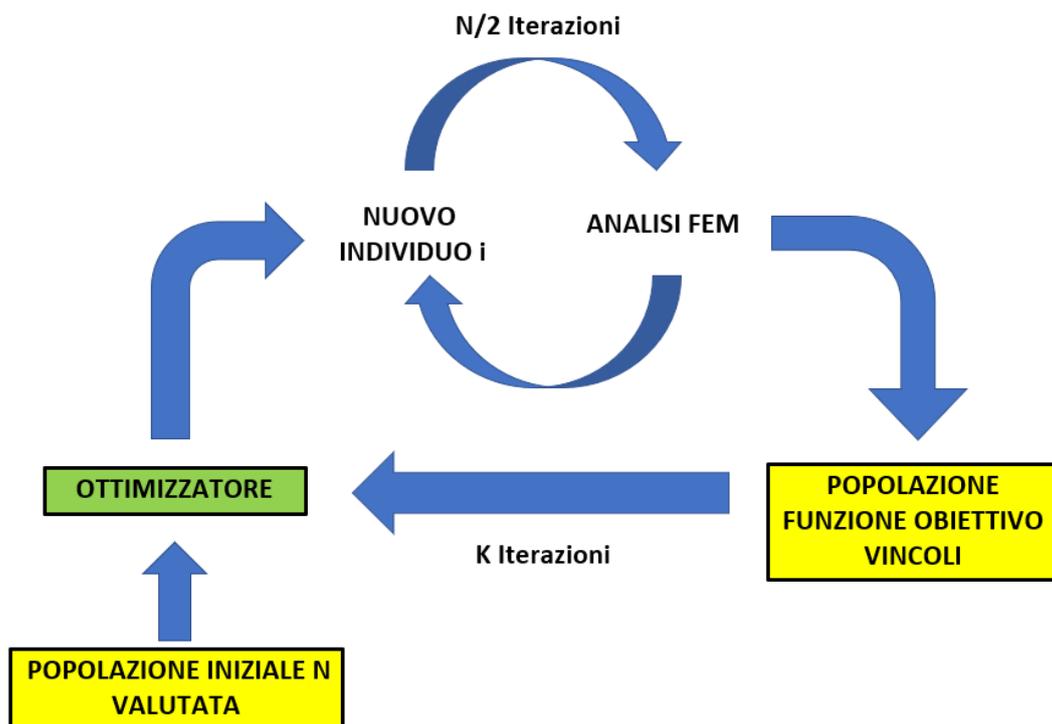


Figura 51 Loop del processo di ottimizzazione: il processo prevede una popolazione iniziale valutata come dato in entrata.  $K$  equivale al numero massimo di iterazioni impostato per l'analisi e  $N$  al numero di individui totale della popolazione.

## 5.4 CASI STUDIO

La struttura presa in esame ha attraversato un processo di ottimizzazione distinguendo due fondamentali tipi di caso studio. In primo luogo, è stata impostata un processo di ottimizzazione fissando la geometria della struttura e impostando un numero crescente di parametri per le caratteristiche meccaniche dei vari gruppi di elementi strutturali che vanno a comporre la struttura. I risultati delle analisi sono poi stati confrontati per vedere l'evolversi del peso della struttura in funzione del numero crescente di parametri. Sono difatti state impostate quattro analisi diverse con tre, cinque, dieci e diciassette parametri strutturali.

Il secondo caso studio è stato impostato analizzando la struttura impostando, oltre ai parametri legati alle caratteristiche meccaniche della struttura, anche dei parametri geometrici al fine di ricavare la forma più performante per la struttura analizzata. I casi studio sono stati impostati sapendo a posteriori quale fosse il possibile risultato dell'analisi così che si potesse verificare il corretto funzionamento del processo di ottimizzazione creato. Sono state quindi impostate due analisi che prendessero in considerazione cinque parametri meccanici e un parametro geometrico e un'analisi con sette parametri meccanici e due parametri geometrici.

*ANALISI PARAMETRICHE*

	Nominativo	PARAMETRI MECCANICI	PARAMETRI GEOMETRICI
CASO STUDIO 1	A	3	0
	B	5	0
	C	10	0
	D	17	0
CASO STUDIO 2	E	5	1
	F	5	1
	G	5	2

### 5.4.1 CASO STUDIO 1

Sono qui riportate le analisi condotte, i risultati e le impostazioni utilizzate per le diverse casistiche analizzate. I profili selezionati e indicizzati sono stati scelti da una lista di 84 profili ordinati in funzione delle dimensioni dell'area. Sono stati scelti esclusivamente profili cavi circolari per il processo di ottimizzazione. È riportato l'elenco delle sezioni con le dimensioni del diametro e lo spessore in metri:

*Tabella profili usati nell'ottimizzazione*

CHS0.4x0.01	CHS0.9x0.026	CHS1x0.048	CHS1.5x0.052
CHS0.5x0.012	CHS0.7x0.034	CHS1.2x0.04	CHS1.4x0.056
CHS0.4x0.016	CHS0.8x0.03	CHS1.1x0.044	CHS1.8x0.046
CHS0.6x0.014	CHS1x0.024	CHS1.5x0.034	CHS1.9x0.044
CHS0.4x0.022	CHS0.7x0.04	CHS1.4x0.038	CHS1.7x0.05
CHS0.5x0.018	CHS0.8x0.036	CHS1.3x0.042	CHS1.6x0.054
CHS0.4x0.028	CHS1.1x0.026	CHS1.1x0.05	CHS1.5x0.058
CHS0.7x0.016	CHS0.9x0.032	CHS1.2x0.046	CHS2x0.046
CHS0.5x0.024	CHS1x0.03	CHS1.6x0.036	CHS1.8x0.052
CHS0.6x0.02	CHS0.8x0.042	CHS1.5x0.04	CHS1.7x0.056
CHS0.4x0.034	CHS1.2x0.028	CHS1.4x0.044	CHS1.9x0.05
CHS0.8x0.018	CHS0.9x0.038	CHS1.2x0.052	CHS1.6x0.06
CHS0.5x0.03	CHS1.1x0.032	CHS1.3x0.048	CHS2x0.052
CHS0.7x0.022	CHS1x0.036	CHS1.7x0.038	CHS1.8x0.058
CHS0.6x0.026	CHS1.3x0.03	CHS1.6x0.042	CHS1.7x0.062
CHS0.5x0.036	CHS0.9x0.044	CHS1.5x0.046	CHS1.9x0.056
CHS0.9x0.02	CHS1.2x0.034	CHS1.3x0.054	CHS1.8x0.064
CHS0.6x0.032	CHS1.1x0.038	CHS1.4x0.05	CHS2x0.058
CHS0.8x0.024	CHS1x0.042	CHS1.8x0.04	CHS1.9x0.062
CHS0.7x0.028	CHS1.4x0.032	CHS1.7x0.044	CHS2x0.064
CHS0.6x0.038	CHS1.3x0.036	CHS1.6x0.048	CHS1.9x0.068

#### 5.4.1.1 Caso A

Il processo di ottimizzazione è stato impostato fissando i seguenti parametri:

- Numero massimo di iterazioni 250
- Numero di individui 50
- Varianza minima  $\sigma_{j,min}=1.5$
- Varianza massima  $\sigma_{j,max} = 12$
- Numero di parametri 4

I parametri meccanici per l'assegnazione dei profili alle aste dei vari gruppi sono state scelte da una lista composta da 84 profili cavi a sezione circolare disposti in funzione dell'area. Il dominio dei parametri è stato quindi definito compreso tra l'indice 1 e l'indice 84.

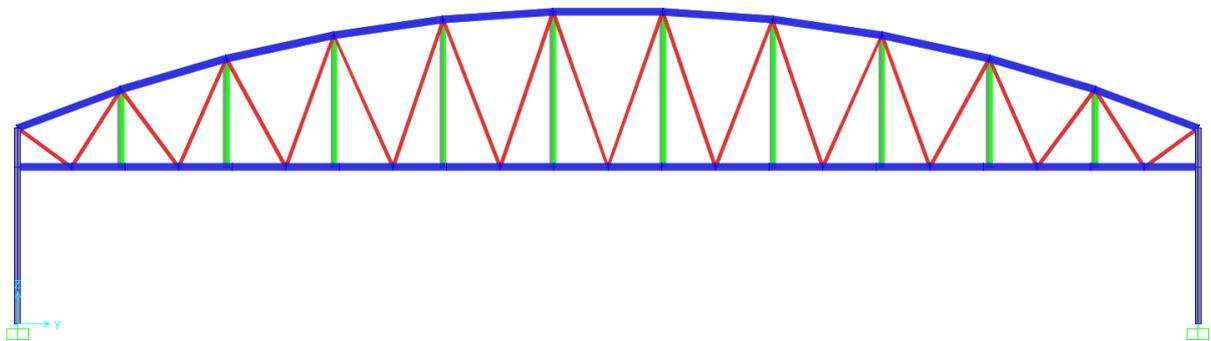


Figura 52 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico.

Sono stati divisi in gruppi i vari elementi strutturali e divisi in sottogruppi per l'assegnazione delle caratteristiche meccaniche a seconda della loro posizione nella travatura reticolare. Alle colonne e la controventatura verticale sono stati assegnate delle sezioni fisse.

Parametri:

- Diagonali 1 parametro.
- Correnti 1 parametro.
- Montanti 1 parametro.

L'analisi è stata condotta con un totale di tre parametri.

Si è visto un andamento decrescente della funzione obiettivo verso una stabilizzazione intorno alla 150-esima iterazione. Dal grafico si può vedere l'andamento della funzione obiettivo al crescere del numero delle iterazioni.



Figura 53 Grafico di andamento della funzione obiettivo.

La soluzione ottimale presenta una massa strutturale pari a 156961 kN. Sono riportate le analisi condotte da Sap2000 per la soluzione ottimale per i vari casi di carico allo stato limite ultimo e allo stato limite di servizio.

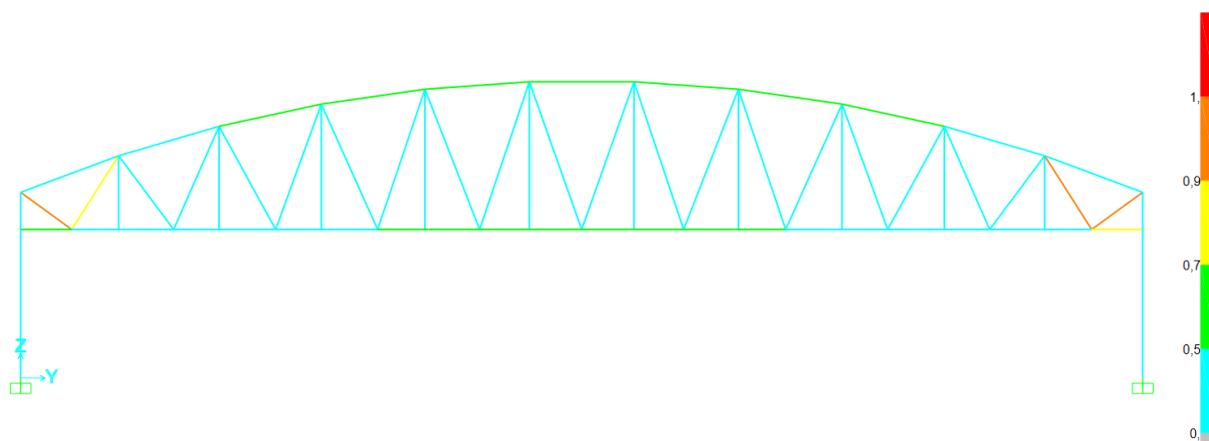


Figura 54 Risultati verifiche strutturali su travatura reticolare vista 2D.

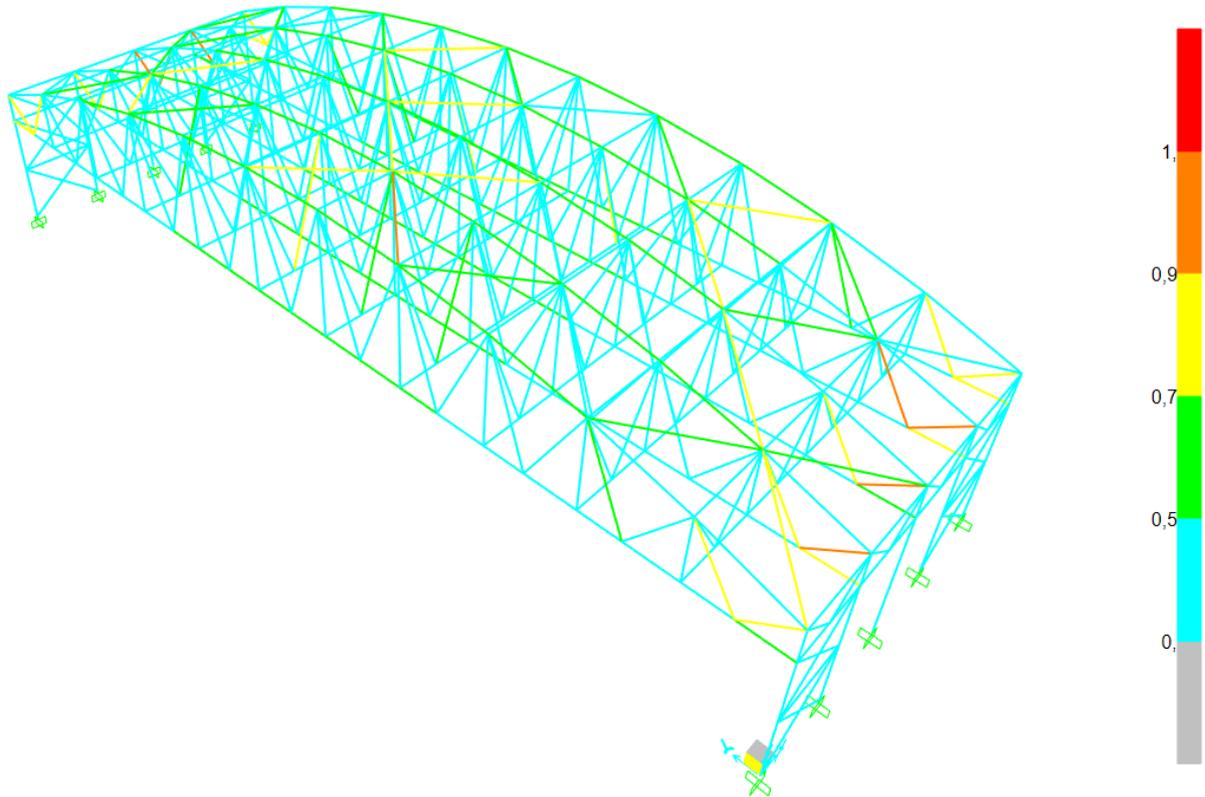


Figura 55 Risultati verifiche strutturali su travatura reticolare vista 3D.

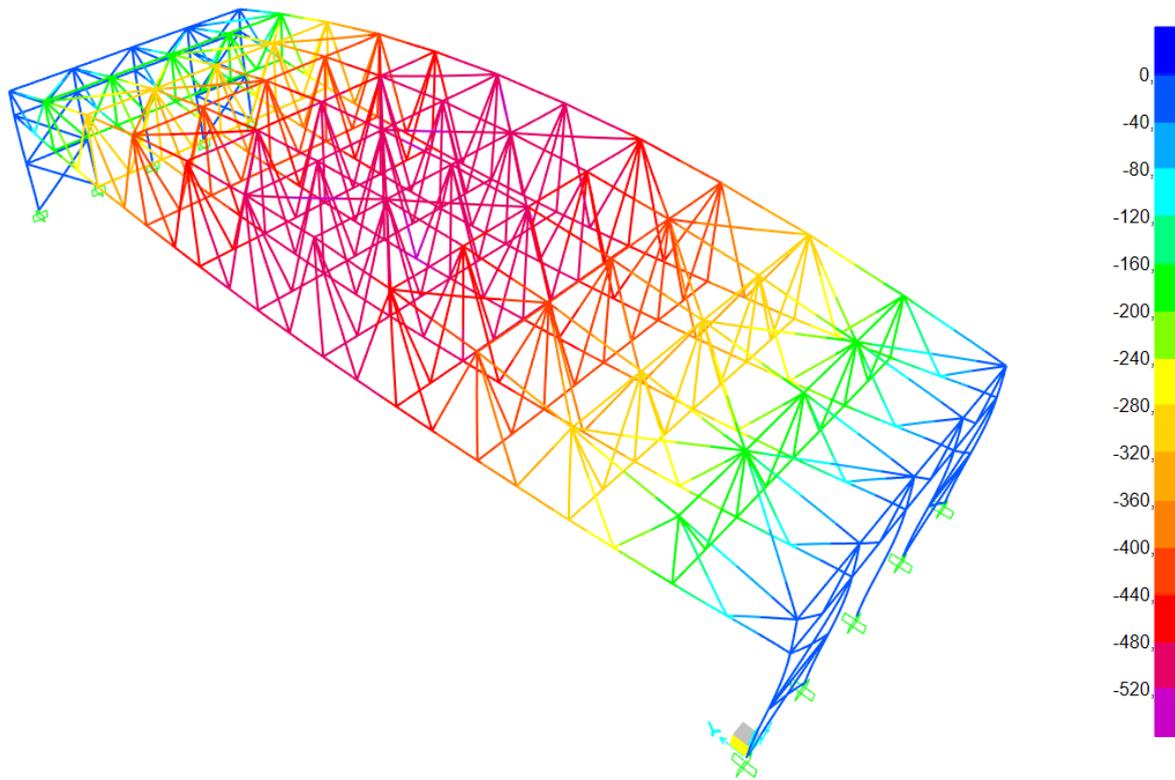
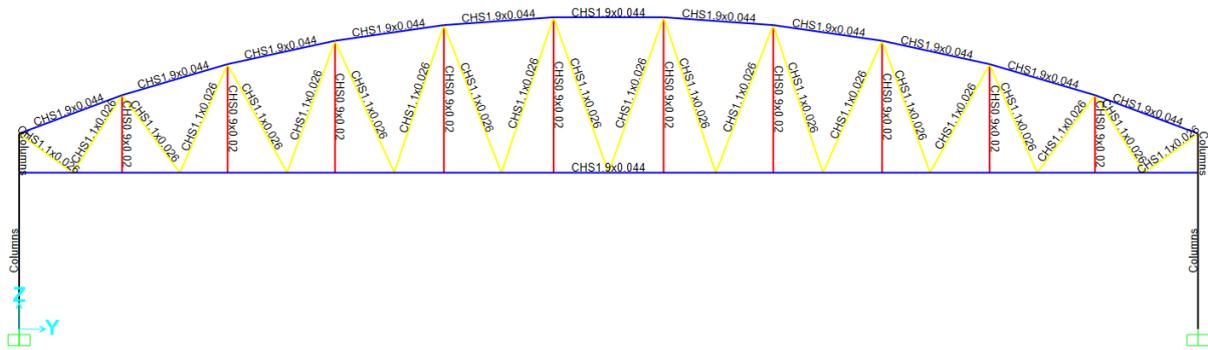


Figura 56 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L.

Sono riportate le soluzioni per i vari parametri impostati.



*Figura 57 Profili risultanti per la soluzione migliore.*

Profili risultanti:

- Diagonali CHS 1100x26 mm.
- Correnti CHS 1900x44mm.
- Montanti CHS 900x20mm.

### 5.4.1.2 Caso B

Il processo di ottimizzazione è stato impostato fissando i seguenti parametri:

- Numero massimo di iterazioni 250
- Numero di individui 50
- Varianza minima  $\sigma_{j,min}=1.5$
- Varianza massima  $\sigma_{j,max} = 12$
- Numero di parametri 5

I parametri meccanici per l'assegnazione dei profili alle aste dei vari gruppi sono state scelte da una lista composta da 84 profili cavi a sezione circolare disposti in funzione dell'area. Il dominio dei parametri è stato quindi definito compreso tra l'indice 1 e l'indice 84.

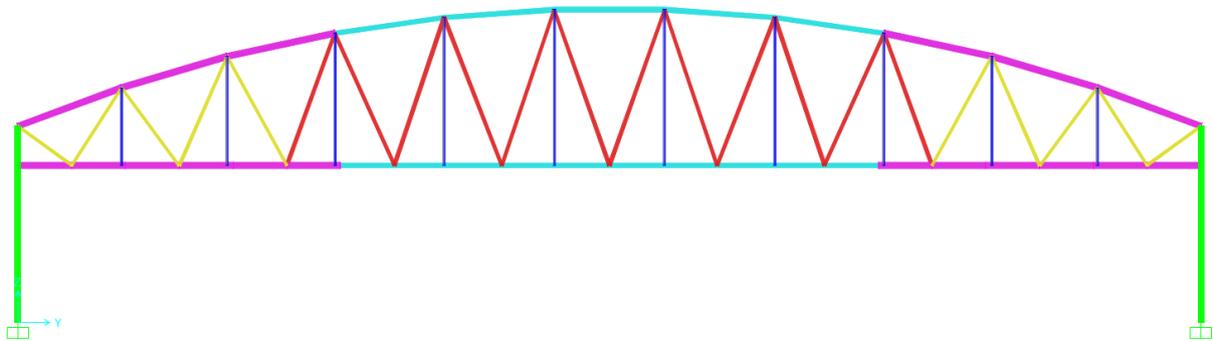


Figura 58 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico.

Sono stati divisi in gruppi i vari elementi strutturali e divisi in sottogruppi per l'assegnazione delle caratteristiche meccaniche a seconda della loro posizione nella travatura reticolare. Alle colonne e la controventatura verticale sono stati assegnate delle sezioni fisse.

Parametri:

- Diagonali 2 parametri.
- Correnti 2 parametri.
- Montanti 1 parametro.

L'analisi è stata condotta con un totale di cinque parametri.

Si è visto un andamento decrescente della funzione obiettivo verso una stabilizzazione intorno alla 125-esima iterazione. Dal grafico si può vedere l'andamento della funzione obiettivo al crescere del numero delle iterazioni.



Figura 59 Grafico di andamento della funzione obiettivo.

La soluzione ottimale presenta una massa strutturale pari a 117205 kN. Sono riportate le analisi condotte da Sap2000 per la soluzione ottimale per i vari casi di carico allo stato limite ultimo e allo stato limite di servizio.

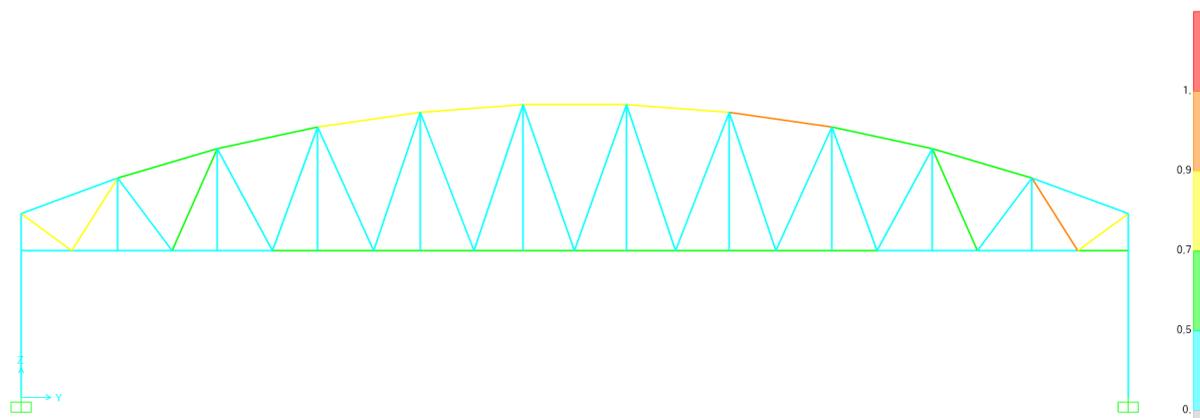


Figura 60 Risultati verifiche strutturali su travatura reticolare vista 2D.

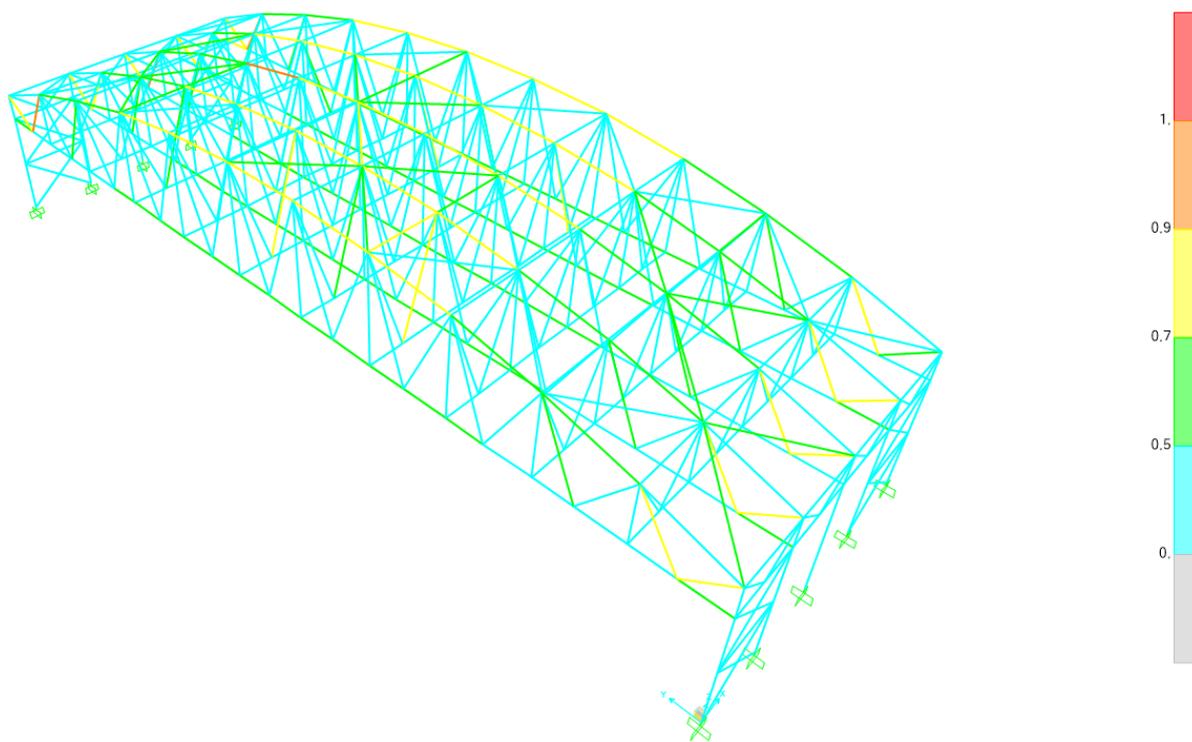


Figura 61 Risultati verifiche strutturali su travatura reticolare vista 3D.

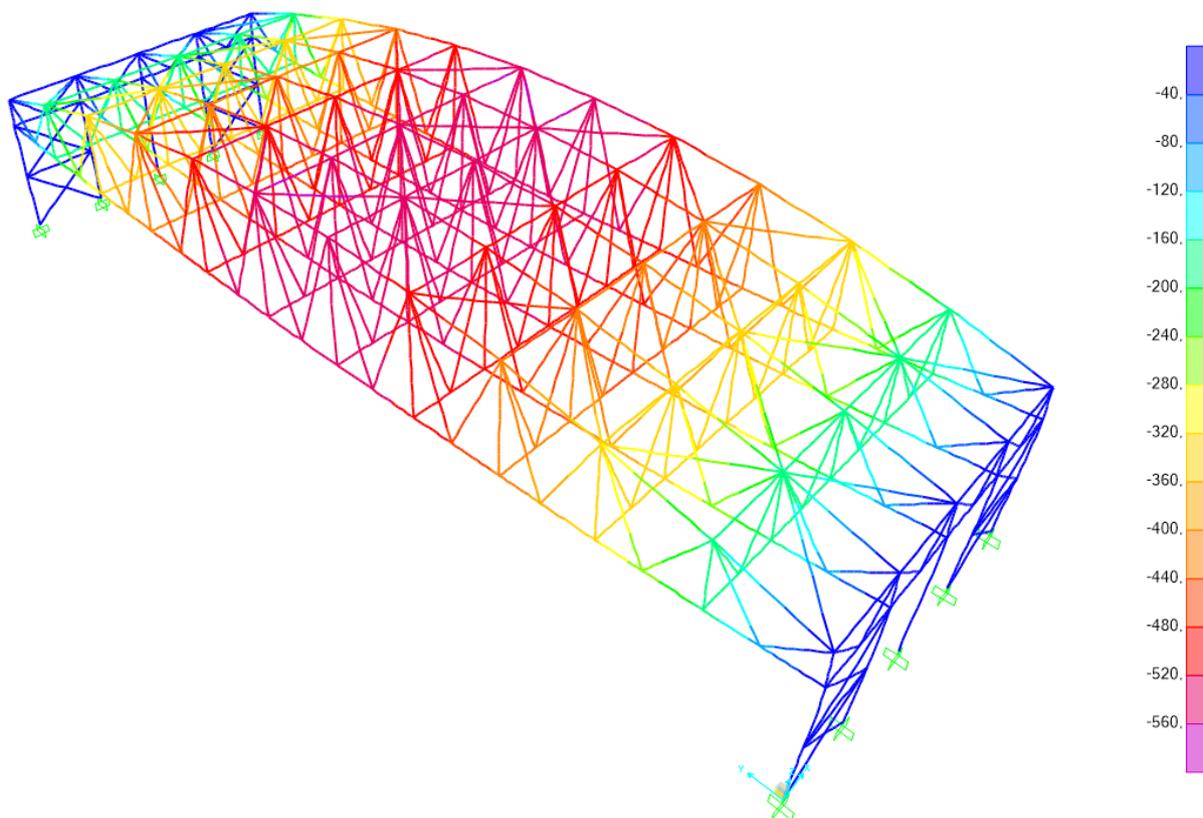
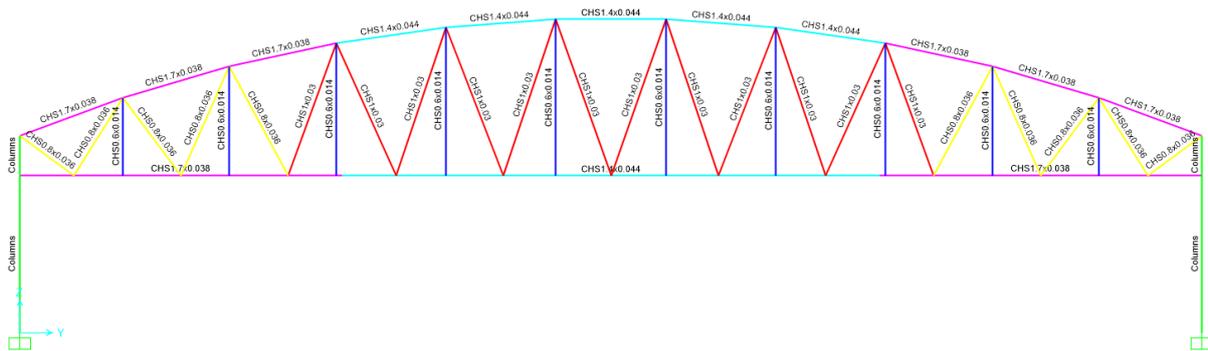


Figura 62 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L.

Sono riportate le soluzioni per i vari parametri impostati.



*Figura 63 Profili risultanti per la soluzione migliore.*

Profili risultanti sono riportati a seconda del gruppo strutturale di appartenenza:

- Diagonali CHS 800x36 mm, CHS 1000x30 mm.
- Correnti CHS 1700x38mm, CHS 1400x44mm.
- Montanti CHS 600x14mm.

### 5.4.1.3 Caso C

Il processo di ottimizzazione è stato impostato fissando i seguenti parametri:

- Numero massimo di iterazioni 250
- Numero di individui 50
- Varianza minima  $\sigma_{j,min}=1.5$
- Varianza massima  $\sigma_{j,max} = 12$
- Numero di parametri 10

I parametri meccanici per l'assegnazione dei profili alle aste dei vari gruppi sono state scelte da una lista composta da 84 profili cavi a sezione circolare disposti in funzione dell'area. Il dominio dei parametri è stato quindi definito compreso tra l'indice 1 e l'indice 84.

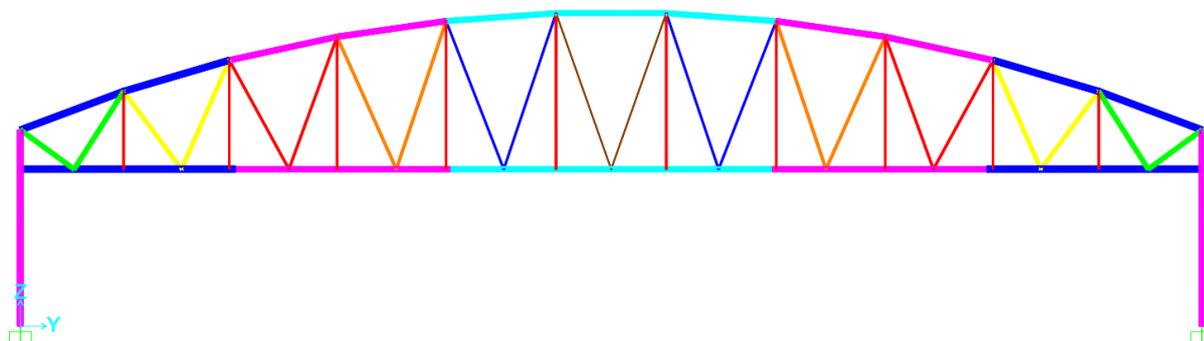


Figura 64 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico.

Sono stati divisi in gruppi i vari elementi strutturali e divisi in sottogruppi per l'assegnazione delle caratteristiche meccaniche a seconda della loro posizione nella travatura reticolare. Alle colonne e la controventatura verticale sono stati assegnate delle sezioni fisse.

Parametri:

- Diagonali 6 parametri.
- Correnti 3 parametri.
- Montanti 1 parametro.

L'analisi è stata condotta con un totale di dieci parametri.

Si è visto un andamento decrescente della funzione obiettivo verso una stabilizzazione intorno alla 190-esima iterazione. Dal grafico si può vedere l'andamento della funzione obiettivo al crescere del numero delle iterazioni.

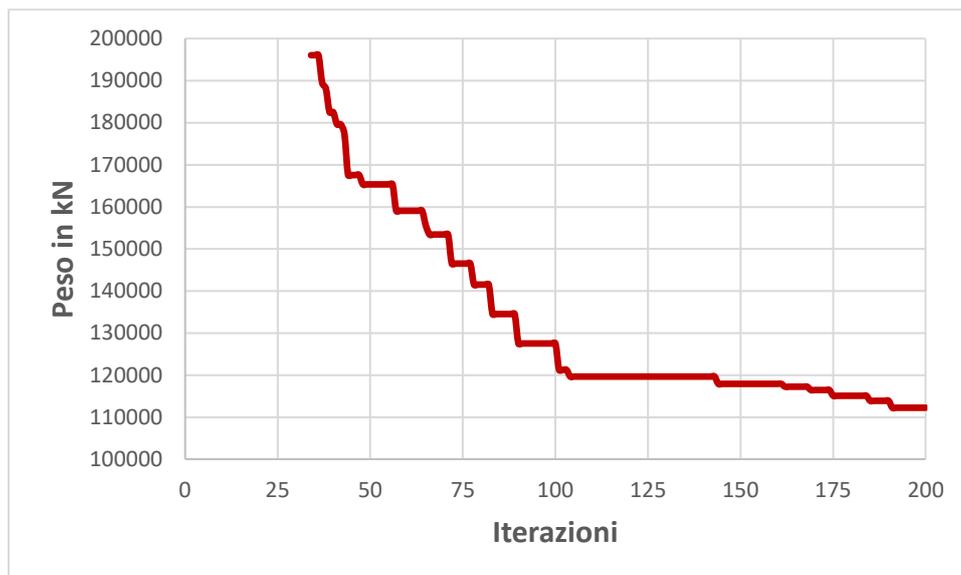


Figura 65 Grafico di andamento della funzione obiettivo.

La soluzione ottimale presenta una massa strutturale pari a 112258 kN. Sono riportate le analisi condotte da Sap2000 per la soluzione ottimale per i vari casi di carico allo stato limite ultimo e allo stato limite di servizio.

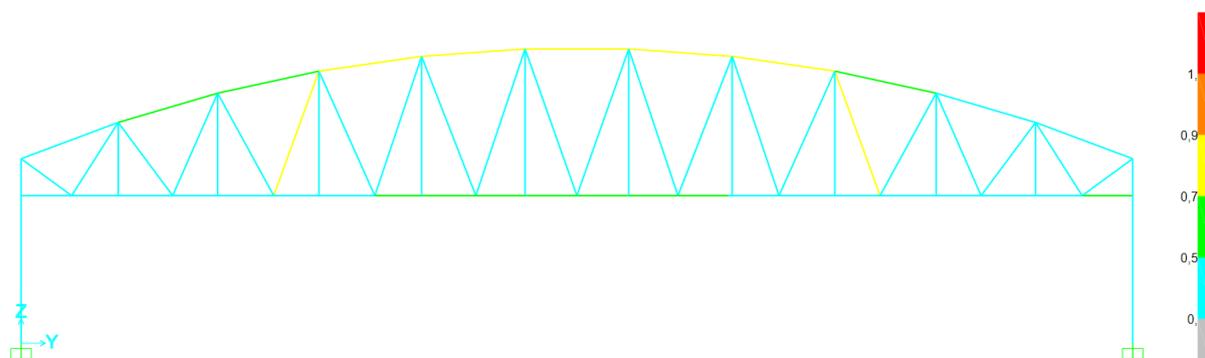


Figura 66 Risultati verifiche strutturali su travatura reticolare vista 2D.

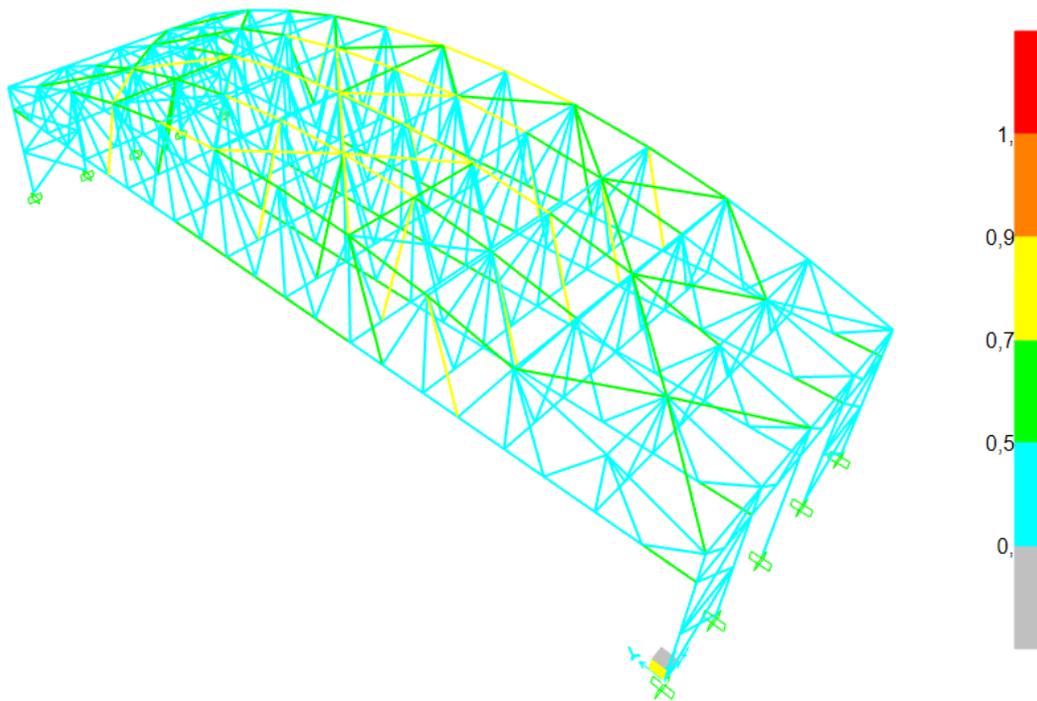


Figura 67 Risultati verifiche strutturali su travatura reticolare vista 3D.

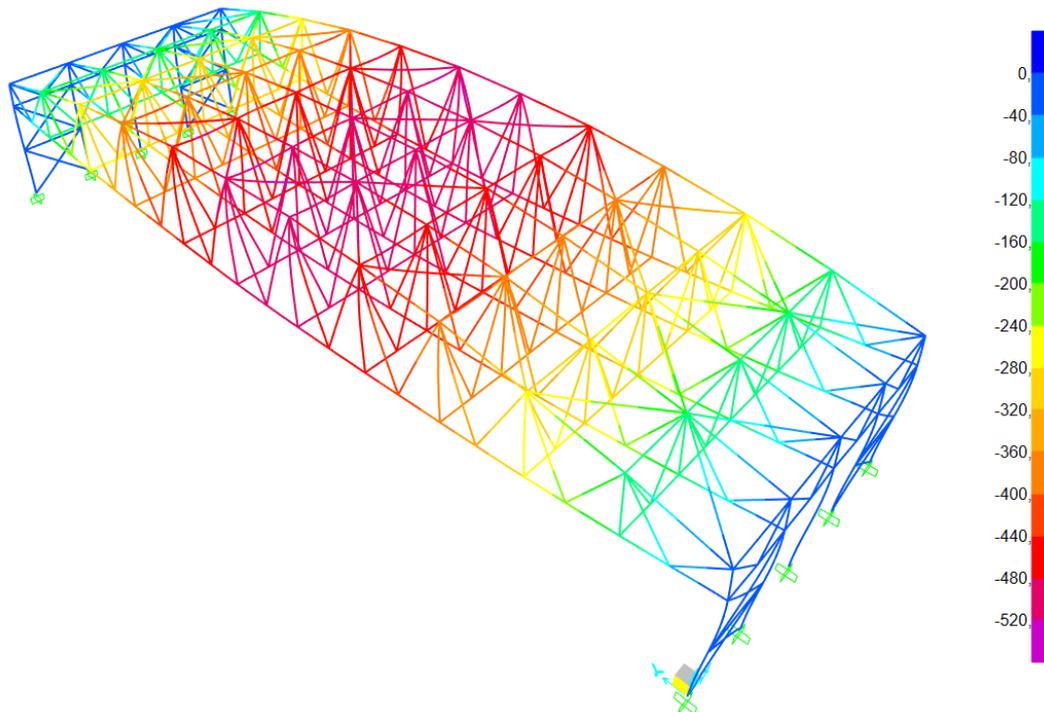


Figura 68 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L.

Sono riportate le soluzioni per i vari parametri impostati.

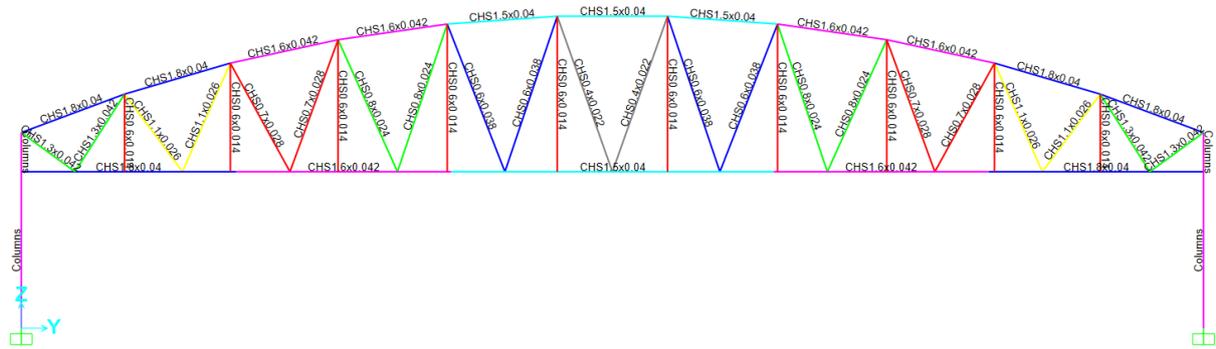


Figura 69 Profili risultanti per la soluzione migliore.

Profili risultanti sono riportati a seconda del gruppo strutturale di appartenenza:

- Diagonali CHS 1300x42 mm, CHS 1100x26 mm, CHS 700x28 mm, CHS 800x24 mm, CHS 600x38 mm, CHS 400x22mm.
- Correnti CHS 1800x40mm, CHS 1600x42mm, CHS 1500x40mm.
- Montanti CHS 600x14mm.

#### 5.4.1.4 Caso D

Il processo di ottimizzazione è stato impostato fissando i seguenti parametri:

- Numero massimo di iterazioni 250
- Numero di individui 50
- Varianza minima  $\sigma_{j,min}=1.5$
- Varianza massima  $\sigma_{j,max} = 12$
- Numero di parametri 18

I parametri meccanici per l'assegnazione dei profili alle aste dei vari gruppi sono state scelte da una lista composta da 84 profili cavi a sezione circolare disposti in funzione dell'area. Il dominio dei parametri è stato quindi definito compreso tra l'indice 1 e l'indice 84.

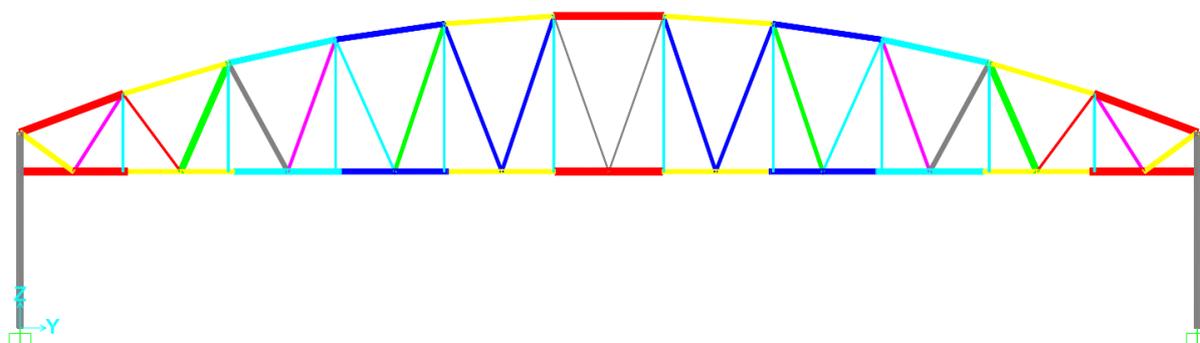


Figura 70 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico.

Sono stati divisi in gruppi i vari elementi strutturali e divisi in sottogruppi per l'assegnazione delle caratteristiche meccaniche a seconda della loro posizione nella travatura reticolare. Alle colonne e la controventatura verticale sono stati assegnate delle sezioni fisse.

Parametri:

- Diagonali 10 parametri.
- Correnti 6 parametri.
- Montanti 1 parametro.

L'analisi è stata condotta con un totale di diciassette parametri.

Si è visto un andamento decrescente della funzione obiettivo verso una stabilizzazione intorno alla 225-esima iterazione. Dal grafico si può vedere l'andamento della funzione obiettivo al crescere del numero delle iterazioni.

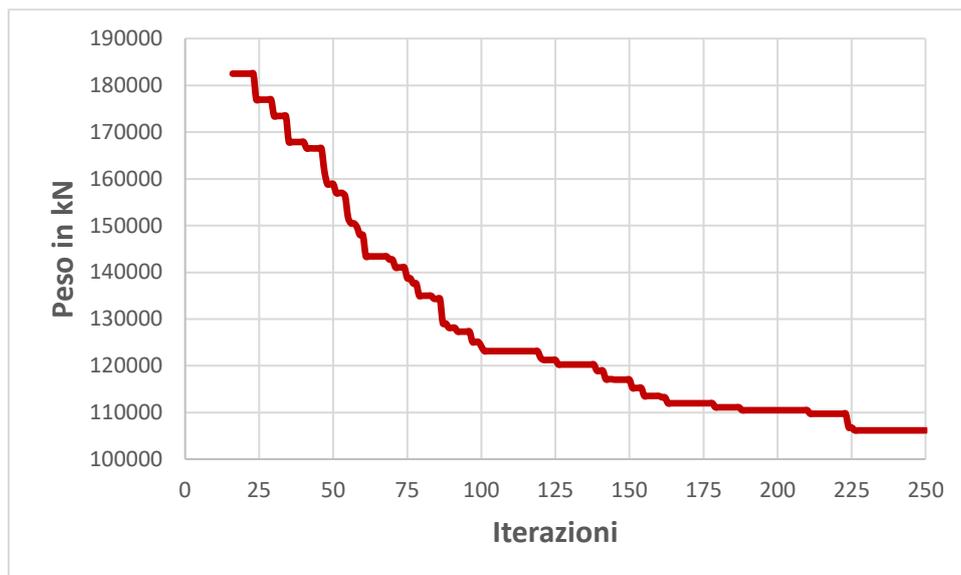


Figura 71 Grafico di andamento della funzione obiettivo.

La soluzione ottimale presenta una massa strutturale pari a 106151 kN. Sono riportate le analisi condotte da Sap2000 per la soluzione ottimale per i vari casi di carico allo stato limite ultimo e allo stato limite di servizio.

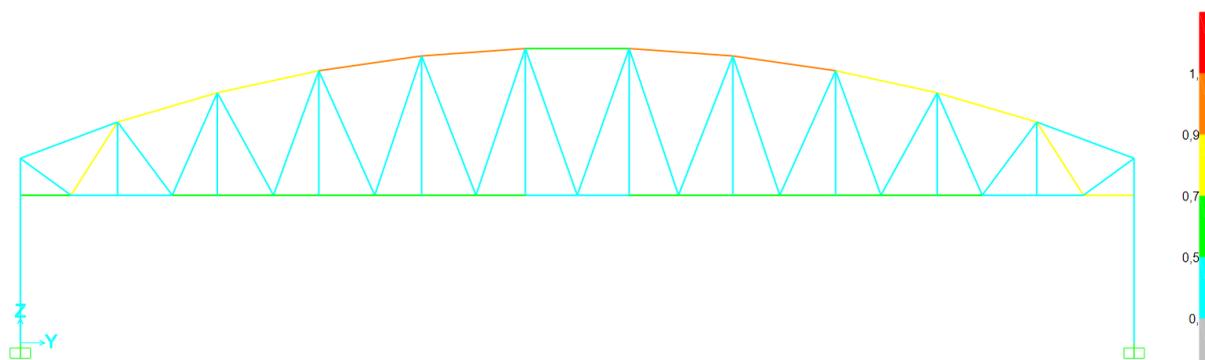


Figura 72 Risultati verifiche strutturali su travatura reticolare vista 2D.

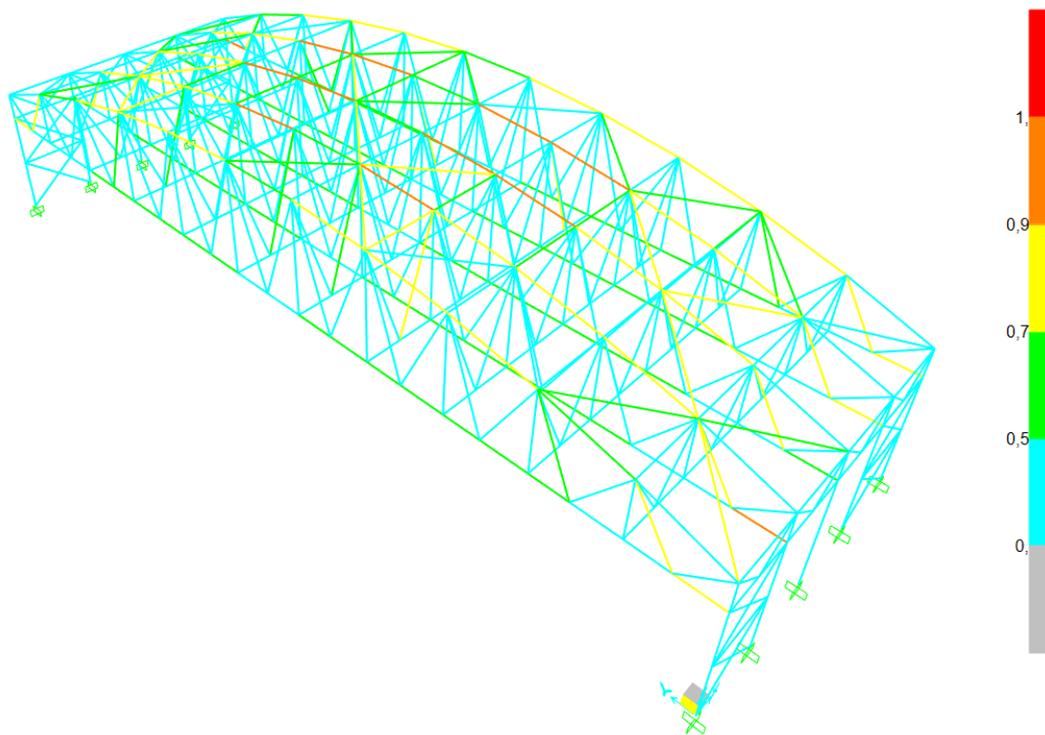


Figura 73 Risultati verifiche strutturali su travatura reticolare vista 3D.

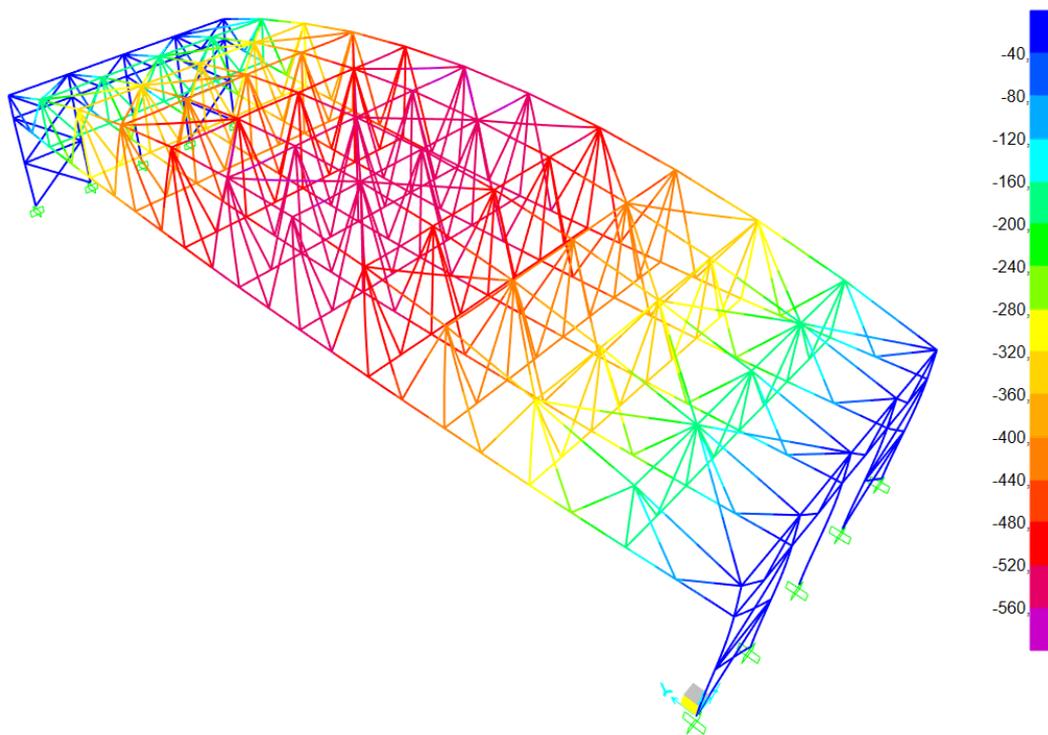


Figura 74 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L.

Sono riportate le soluzioni per i vari parametri impostati.

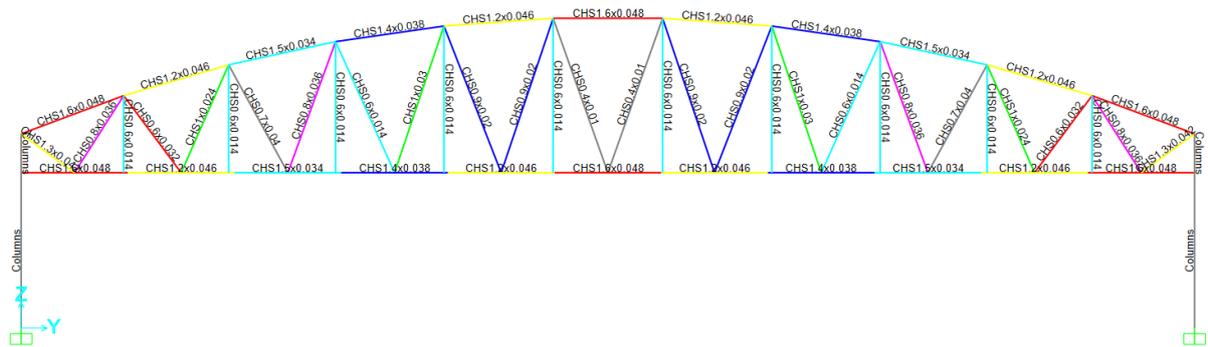


Figura 75 Profili risultanti per la soluzione migliore.

Profili risultanti sono riportati a seconda del gruppo strutturale di appartenenza:

- Diagonali CHS 1300x40 mm, CHS 800x36 mm, CHS 600x32 mm, CHS 1000x24 mm, CHS 700x40 mm, CHS 800x36mm, CHS 600x14mm, CHS 1000x30mm, CHS 900x20mm, CHS 400x10mm.
- Correnti CHS 1600x48mm, CHS 1200x42mm, CHS 1500x34mm, CHS 1400x38mm, CHS 1200x46mm, CHS 1600x48mm.
- Montanti CHS 600x14mm.

## 5.4.2 CASO STUDIO 2

Sono qui riportate le analisi condotte, i risultati e le impostazioni utilizzate per le diverse casistiche analizzate. Nei vari casi sono stati imposti come parametro geometrico variabile la curvatura dei correnti superiori e inferiori. È stato scelto di impostare la variazione del parametro geometrico in funzione della distanza dalla corda inferiore, passante per gli estremi dei correnti, e il punto in mezzera del corrente

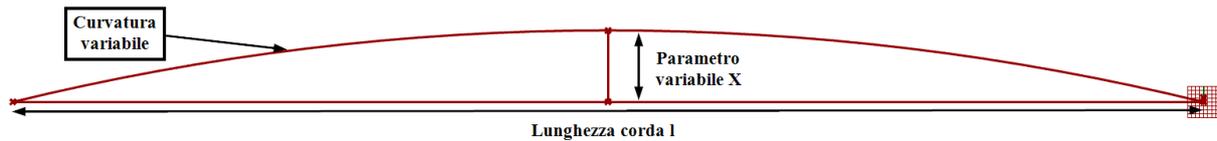


Figura 76 Impostazioni dei parametri geometrici per la variazione della curvatura sui correnti.

È riportato il calcolo della curvatura del corrente in funzione della variabile  $x$  e della variabile  $l$ :

$$\frac{1}{r} = \frac{2x}{\frac{l^2}{4} + x^2}$$

Dove  $l$  è la lunghezza della corda inferiore, nel caso della struttura presa in esame la lunghezza della campata fissata a 300 metri e  $x$  è la distanza dalla corda inferiore considerata variabile da un minimo di 0 a un massimo di 30 metri.

### 5.4.2.1 Caso E

Il processo di ottimizzazione è stato impostato fissando i seguenti parametri:

- Numero massimo di iterazioni 250
- Numero di individui 50
- Varianza minima  $\sigma_{j,min}=1.5$
- Varianza massima  $\sigma_{j,max} = 12$
- Numero di parametri totale 6
- Numero parametri meccanici 5
- Numero parametri geometrici 1

I parametri meccanici per l'assegnazione dei profili alle aste dei vari gruppi sono state scelte da una lista composta da 84 profili cavi a sezione circolare disposti in funzione dell'area. Il dominio dei parametri è stato quindi definito compreso tra l'indice 1 e l'indice 84.

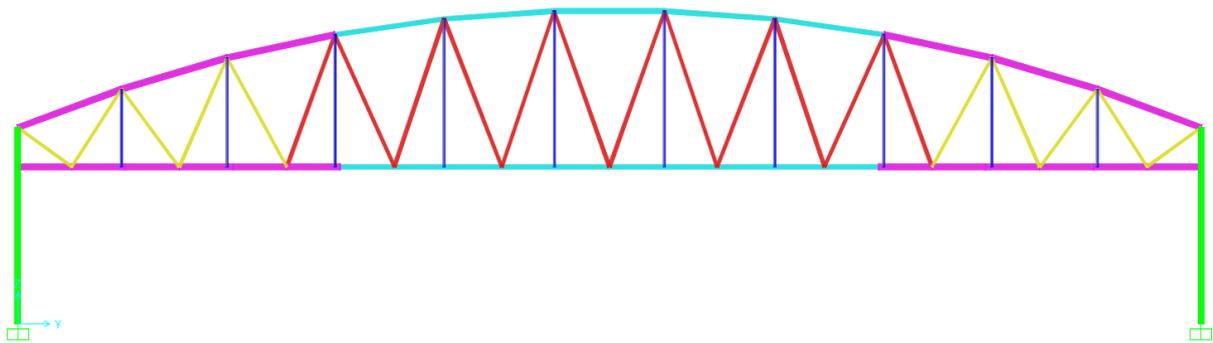


Figura 77 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico.

Sono stati divisi in gruppi i vari elementi strutturali e divisi in sottogruppi per l'assegnazione delle caratteristiche meccaniche a seconda della loro posizione nella travatura reticolare. Alle colonne e la controventatura verticale sono stati assegnate delle sezioni fisse.

È stato impostato come parametro geometrico variabile la curvatura del corrente superiore facendo variare il parametro da un minimo di 0 a un massimo di 30m.

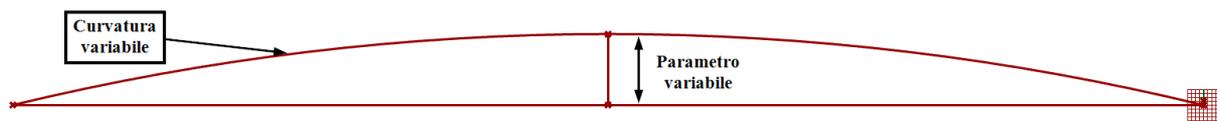


Figura 78 Parametrizzazione geometrica curvatura corrente superiore.

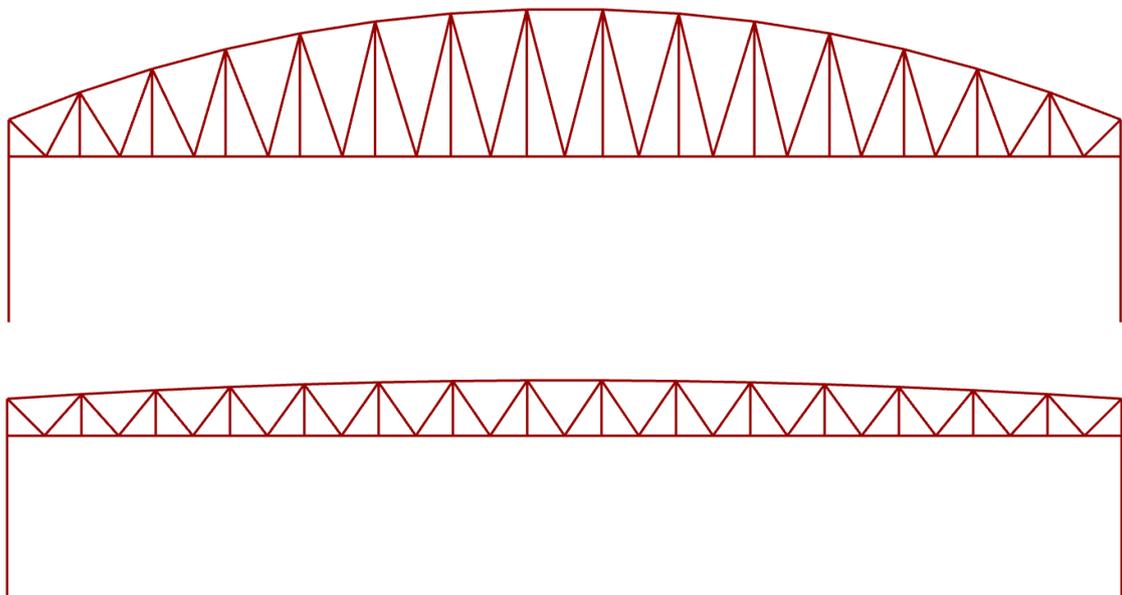


Figura 79 Possibili configurazioni geometriche.

Parametri meccanici:

- Diagonali 2 parametri.
- Correnti 2 parametri.
- Montanti 1 parametro.

Parametri geometrici:

- Curvatura corrente superiore 1 parametro.

L'analisi è stata condotta con un totale di sei parametri.

Si è visto un andamento decrescente della funzione obiettivo verso una stabilizzazione intorno alla 100-esima iterazione. Dal grafico si può vedere l'andamento della funzione obiettivo al crescere del numero delle iterazioni.

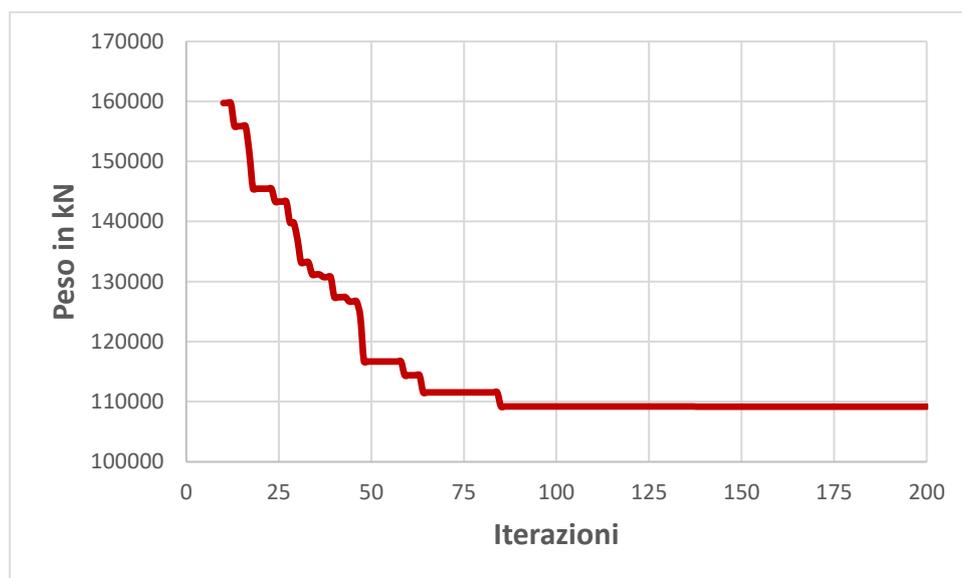


Figura 80 Grafico di andamento della funzione obiettivo.

La soluzione ottimale presenta una massa strutturale pari a 109142 kN. Sono riportate le analisi condotte da Sap2000 per la soluzione ottimale per i vari casi di carico allo stato limite ultimo e allo stato limite di servizio.

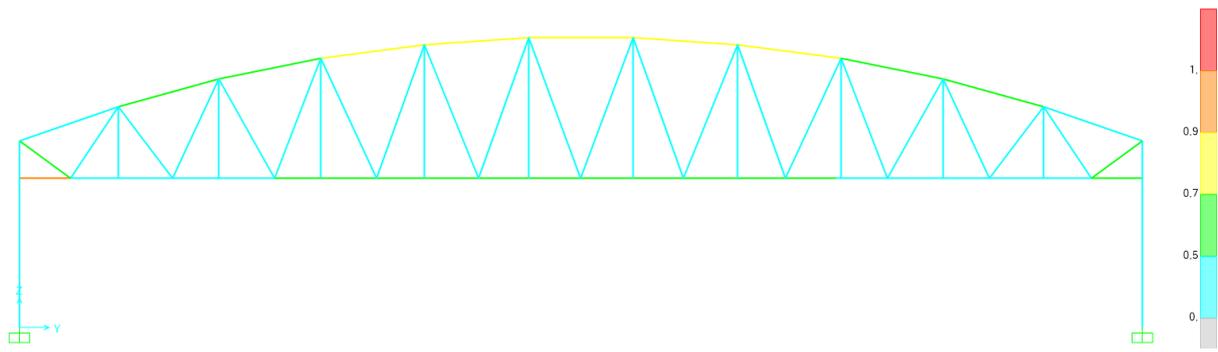


Figura 81 Risultati verifiche strutturali su travatura reticolare vista 2D.

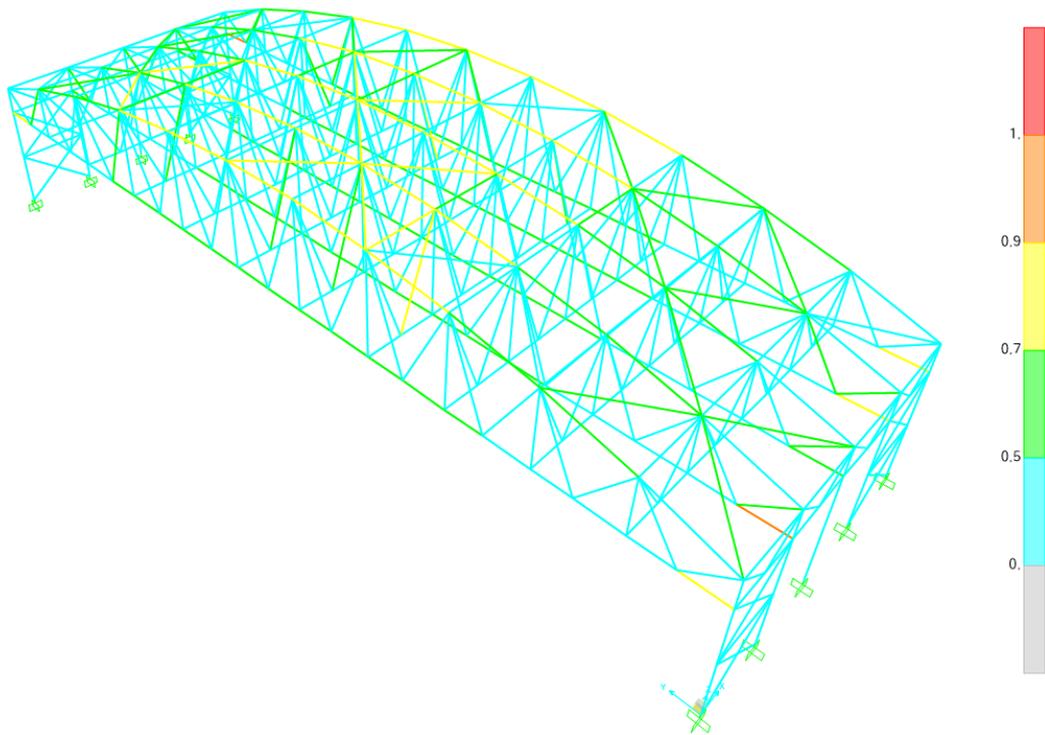


Figura 82 Risultati verifiche strutturali su travatura reticolare vista 3D.

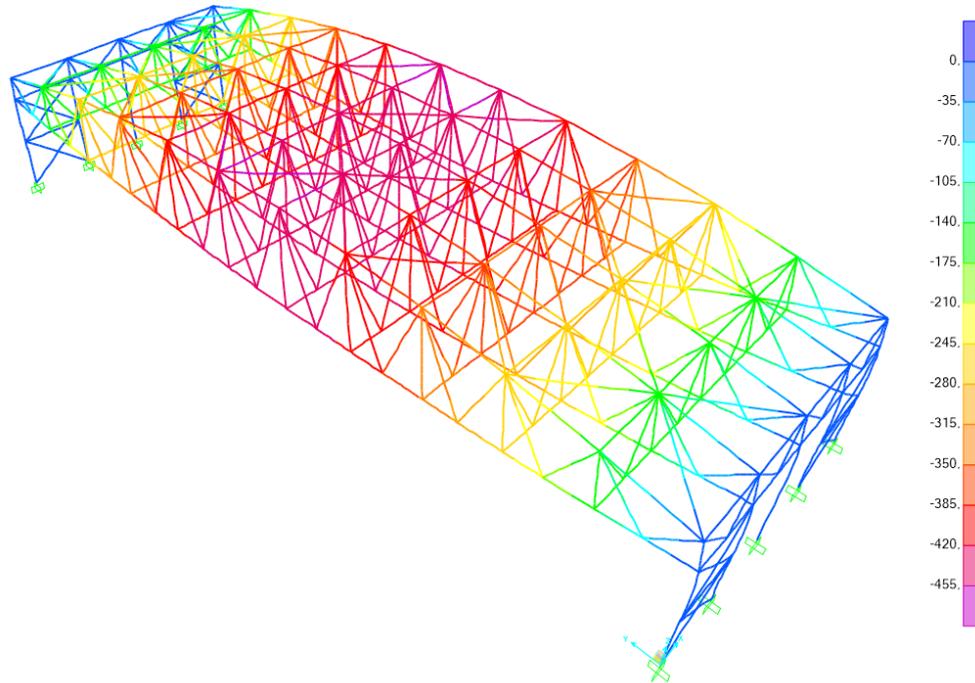


Figura 83 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L.

Sono riportate le soluzioni per i vari parametri impostati.

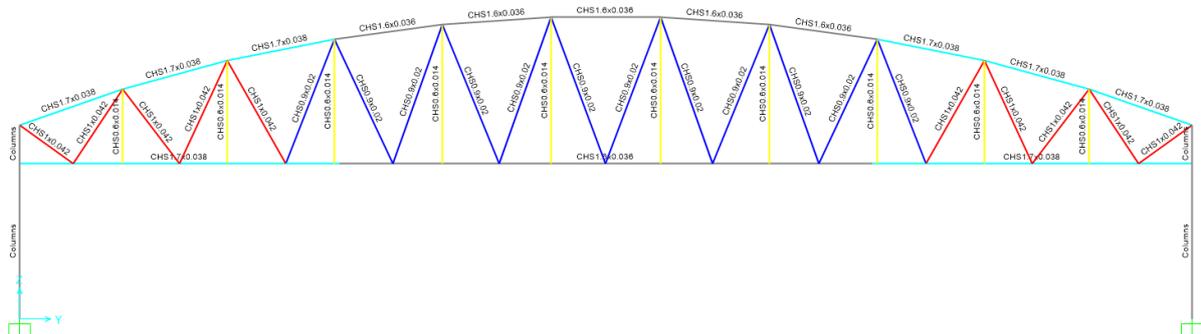


Figura 84 Profili risultanti per la soluzione migliore.

Profili risultanti sono riportati a seconda del gruppo strutturale di appartenenza:

- Diagonali CHS 1000x42 mm, CHS 900x20 mm.
- Correnti CHS 1700x38mm, CHS 1600x36mm.
- Montanti CHS 600x14mm.

Parametri geometrici risultati:

- Distanza dalla corda 28m.

È risultata una curvatura del corrente superiore pari a  $0.056 \frac{1}{m}$ .

### 5.4.2.2 Caso F

Il processo di ottimizzazione è stato impostato fissando i seguenti parametri:

- Numero massimo di iterazioni 250
- Numero di individui 50
- Varianza minima  $\sigma_{j,min}=1.5$
- Varianza massima  $\sigma_{j,max} = 12$
- Numero di parametri totale 6
- Numero parametri meccanici 5
- Numero parametri geometrici 1

I parametri meccanici per l'assegnazione dei profili alle aste dei vari gruppi sono state scelte da una lista composta da 84 profili cavi a sezione circolare disposti in funzione dell'area. Il dominio dei parametri è stato quindi definito compreso tra l'indice 1 e l'indice 84.

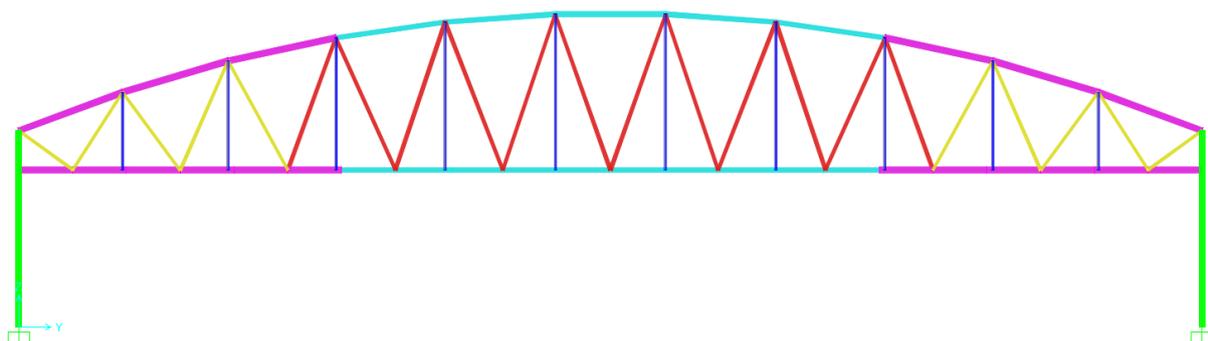


Figura 85 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico.

Sono stati divisi in gruppi i vari elementi strutturali e divisi in sottogruppi per l'assegnazione delle caratteristiche meccaniche a seconda della loro posizione nella travatura reticolare. Alle colonne e la controventatura verticale sono stati assegnate delle sezioni fisse.

È stato impostato come parametro geometrico variabile la curvatura del corrente inferiore facendo variare il parametro da un minimo di 0 a un massimo di 30m.

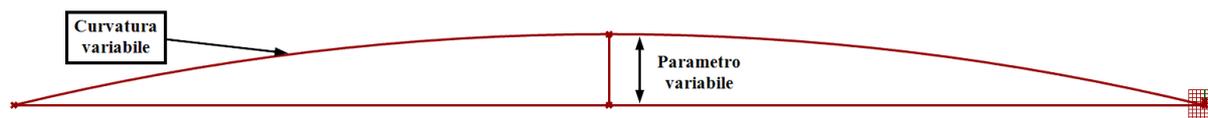
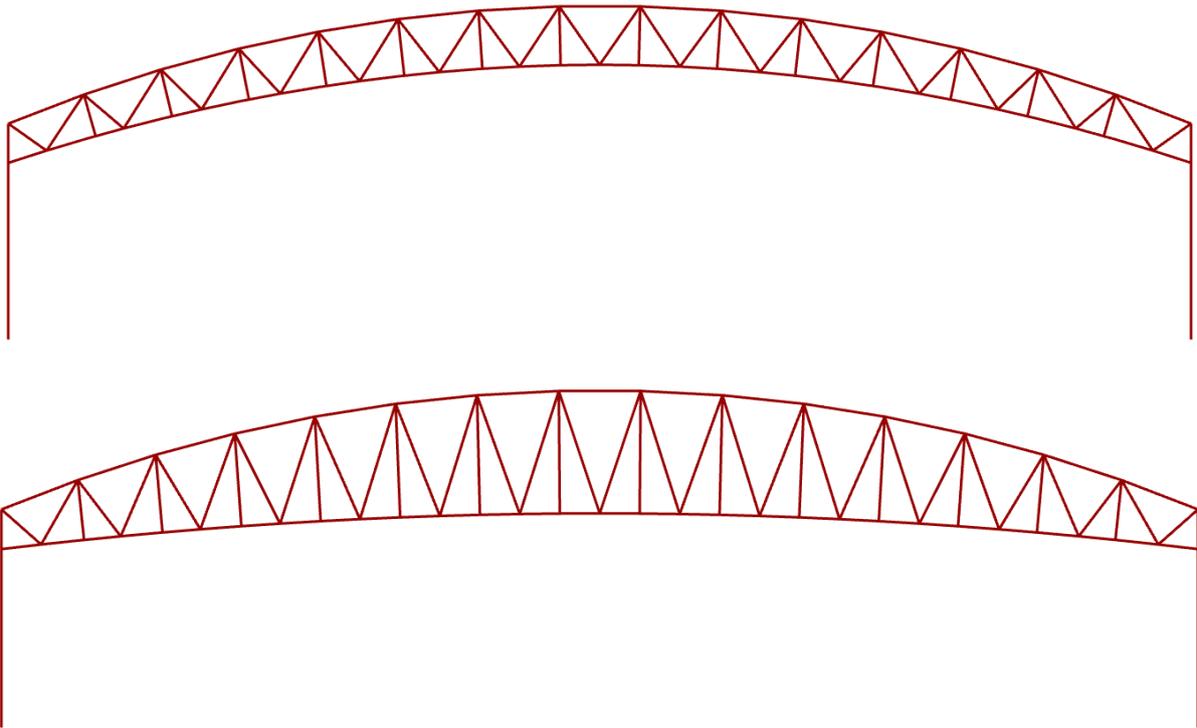


Figura 86 Parametrizzazione geometrica curvatura corrente inferiore.



*Figura 87 Possibili configurazioni geometriche.*

Parametri meccanici:

- Diagonali 2 parametri.
- Correnti 2 parametri.
- Montanti 1 parametro.

Parametri geometrici:

- Curvatura corrente inferiore 1 parametro.

L'analisi è stata condotta con un totale di sei parametri.

Si è visto un andamento decrescente della funzione obiettivo verso una stabilizzazione intorno alla 125-esima iterazione. Dal grafico si può vedere l'andamento della funzione obiettivo al crescere del numero delle iterazioni.

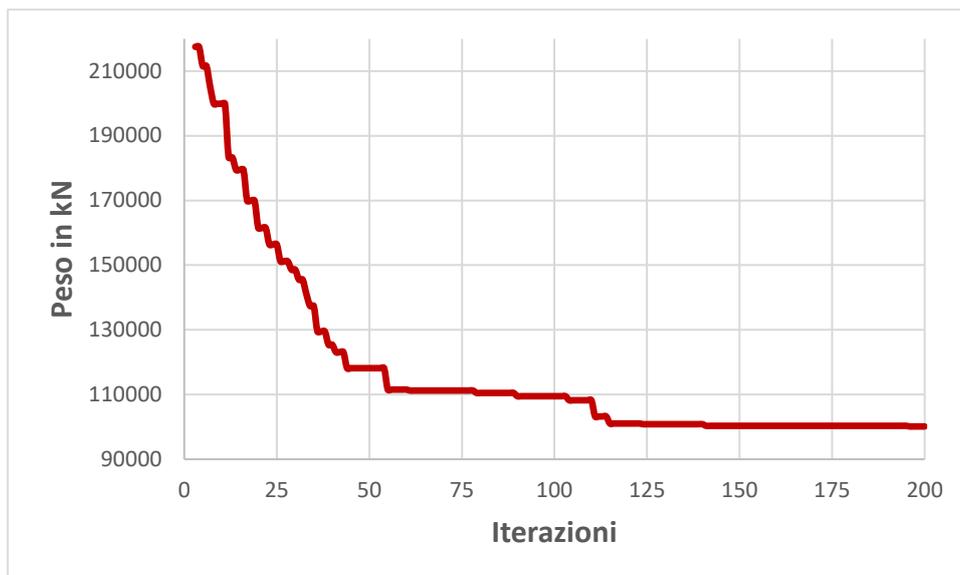


Figura 88 Grafico di andamento della funzione obiettivo.

La soluzione ottimale presenta una massa strutturale pari a 100121 kN. Sono riportate le analisi condotte da Sap2000 per la soluzione ottimale per i vari casi di carico allo stato limite ultimo e allo stato limite di servizio.

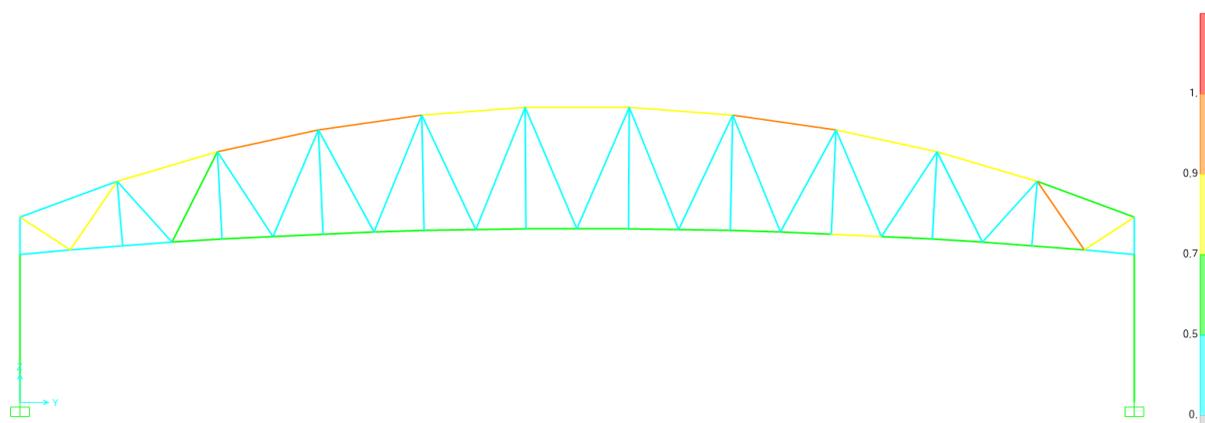


Figura 89 Risultati verifiche strutturali su travatura reticolare vista 2D.

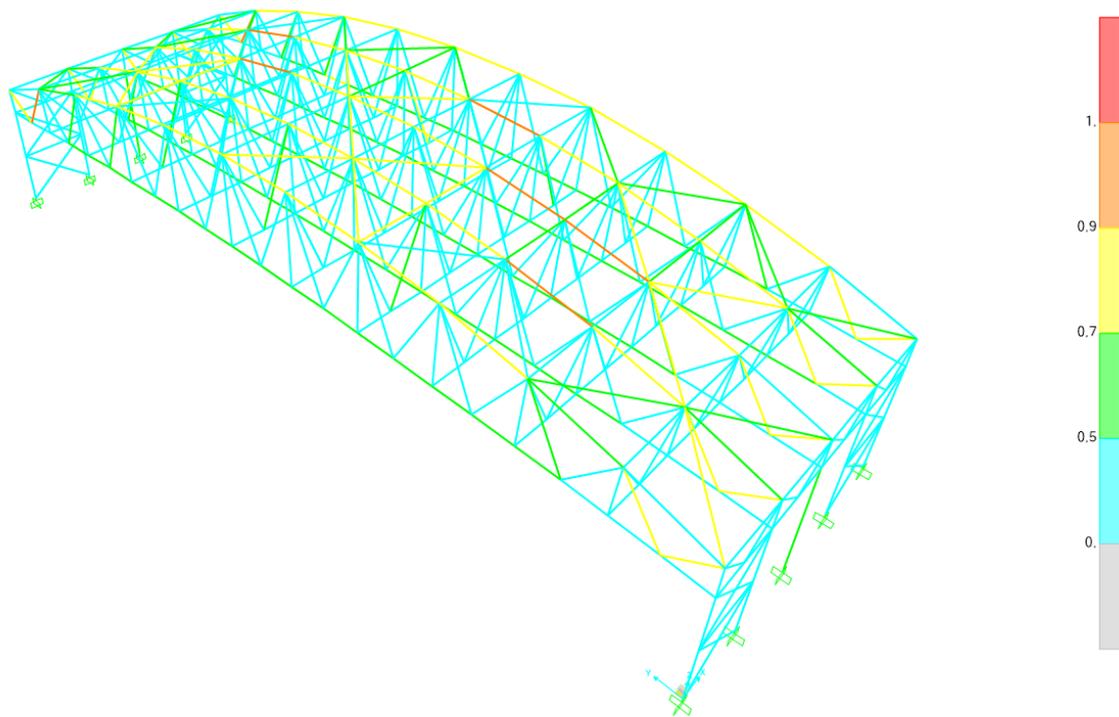


Figura 90 Risultati verifiche strutturali su travatura reticolare vista 3D.

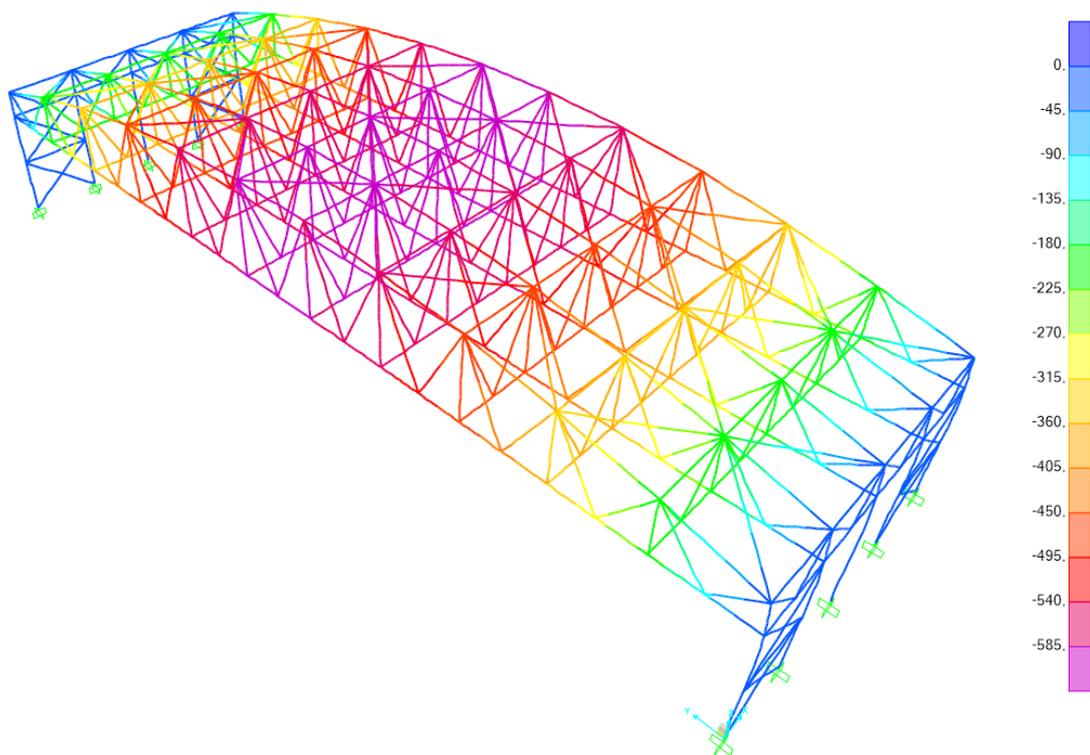


Figura 91 Deformata della travatura reticolare sotto il caso di carico 1.0D+1.0L.

Sono riportate le soluzioni per i vari parametri impostati.

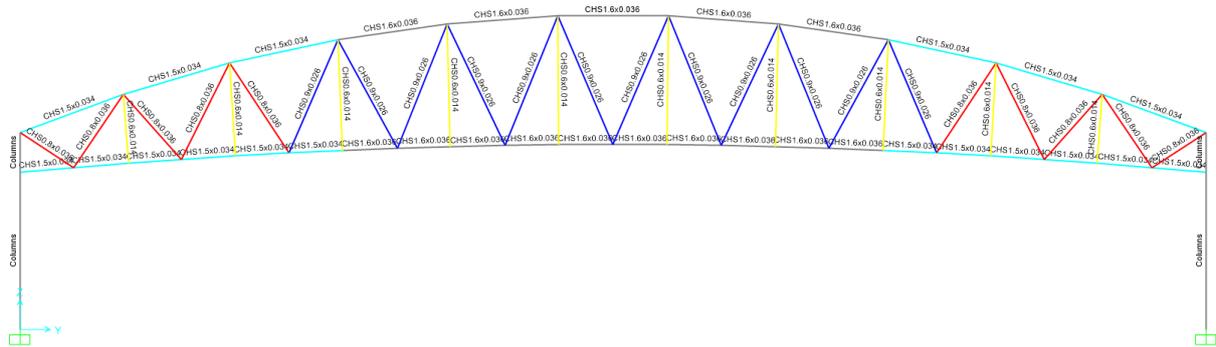


Figura 92 Profili risultanti per la soluzione migliore.

Profili risultanti sono riportati a seconda del gruppo strutturale di appartenenza:

- Diagonali CHS 800x36 mm, CHS 900x26 mm.
- Correnti CHS 1500x34mm, CHS 1600x36mm.
- Montanti CHS 600x14mm.

Parametri geometrici risultati:

- Distanza dalla corda 23m.

È risultata una curvatura del corrente superiore pari a  $0.061 \frac{1}{m}$ .

### 5.4.2.3 Caso G

Il processo di ottimizzazione è stato impostato fissando i seguenti parametri:

- Numero massimo di iterazioni 250
- Numero di individui 50
- Varianza minima  $\sigma_{j,min}=1.5$
- Varianza massima  $\sigma_{j,max} = 12$
- Numero di parametri totale 6
- Numero parametri meccanici 5
- Numero parametri geometrici 2

I parametri meccanici per l'assegnazione dei profili alle aste dei vari gruppi sono state scelte da una lista composta da 84 profili cavi a sezione circolare disposti in funzione dell'area. Il dominio dei parametri è stato quindi definito compreso tra l'indice 1 e l'indice 84.

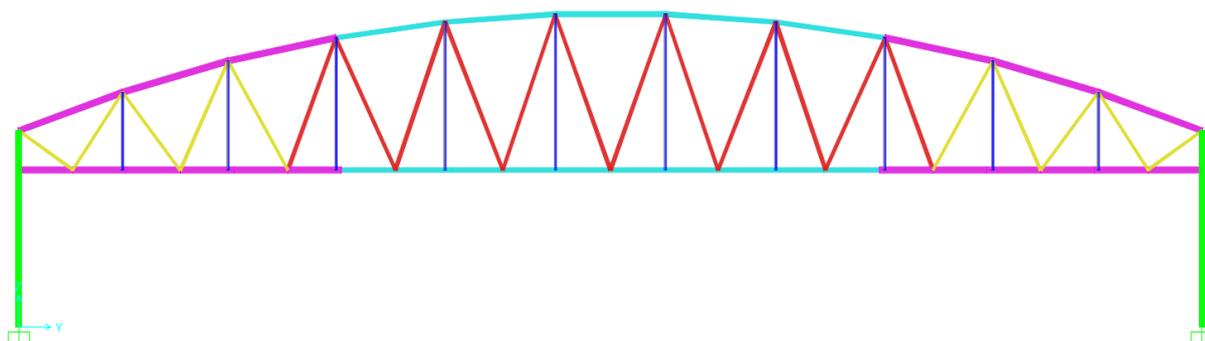


Figura 93 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico.

Sono stati divisi in gruppi i vari elementi strutturali e divisi in sottogruppi per l'assegnazione delle caratteristiche meccaniche a seconda della loro posizione nella travatura reticolare. Alle colonne e la controventatura verticale sono stati assegnate delle sezioni fisse.

È stato impostato come primo parametro geometrico variabile la curvatura del corrente inferiore facendo variare il parametro da un minimo di -30m a un massimo di 30m.

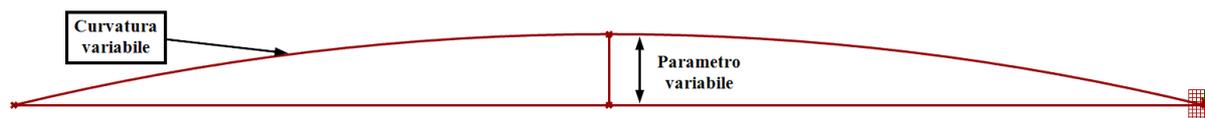
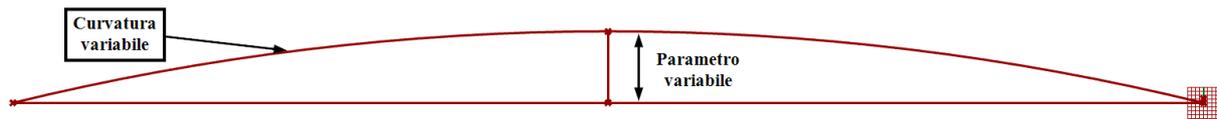
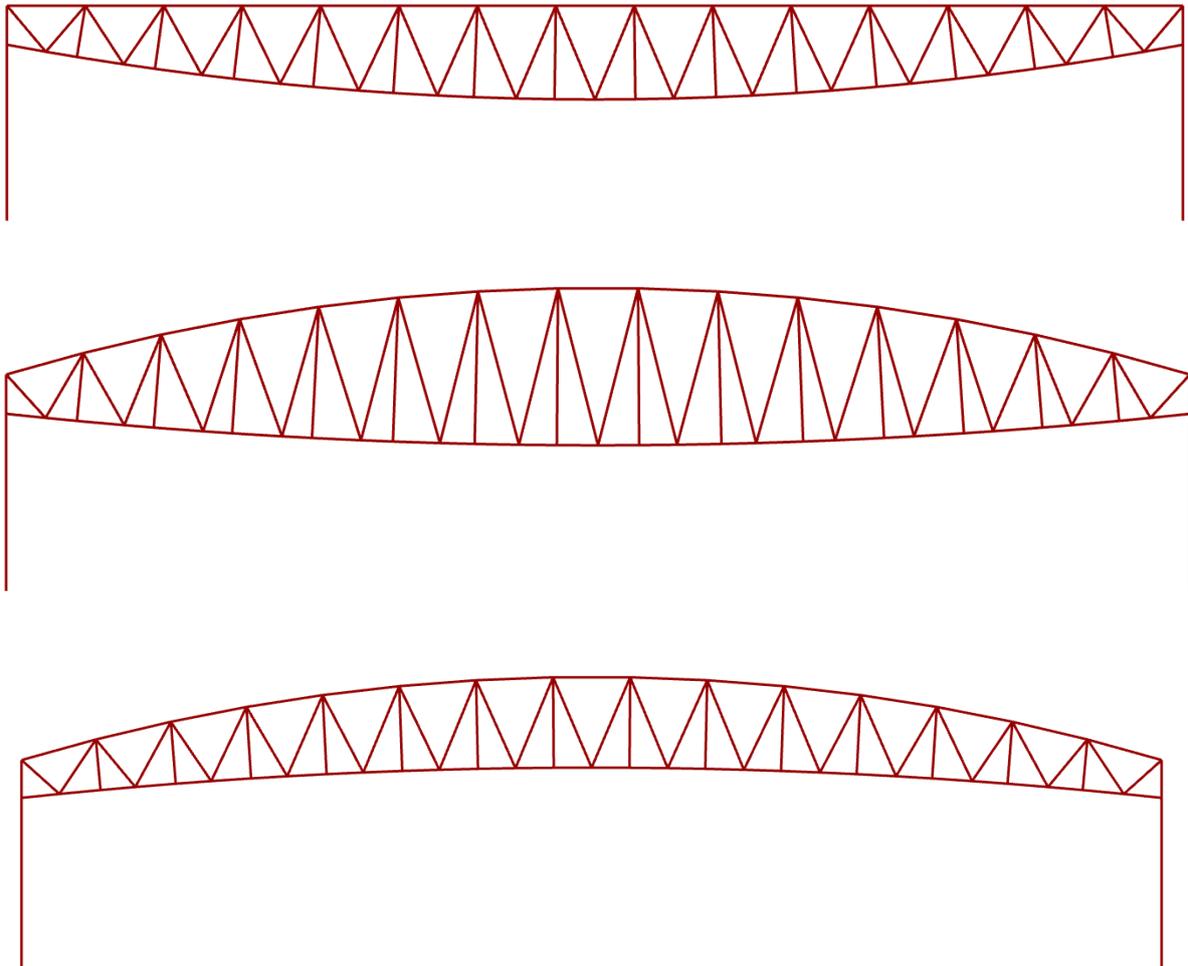


Figura 94 Parametrizzazione geometrica curvatura corrente inferiore.

È stato impostato come secondo parametro geometrico variabile la curvatura del corrente superiore facendo variare il parametro da un minimo di 0 a un massimo di 30m.



*Figura 95 Parametrizzazione geometrica curvatura corrente superiore.*



*Figura 96 Possibili configurazioni geometriche.*

Parametri meccanici:

- Diagonali 2 parametri.
- Correnti 2 parametri.
- Montanti 1 parametro.

Parametri geometrici:

- Curvatura corrente inferiore 1 parametro.

L'analisi è stata condotta con un totale di sei parametri.

Si è visto un andamento decrescente della funzione obiettivo verso una stabilizzazione intorno alla 150-esima iterazione. Dal grafico si può vedere l'andamento della funzione obiettivo al crescere del numero delle iterazioni.

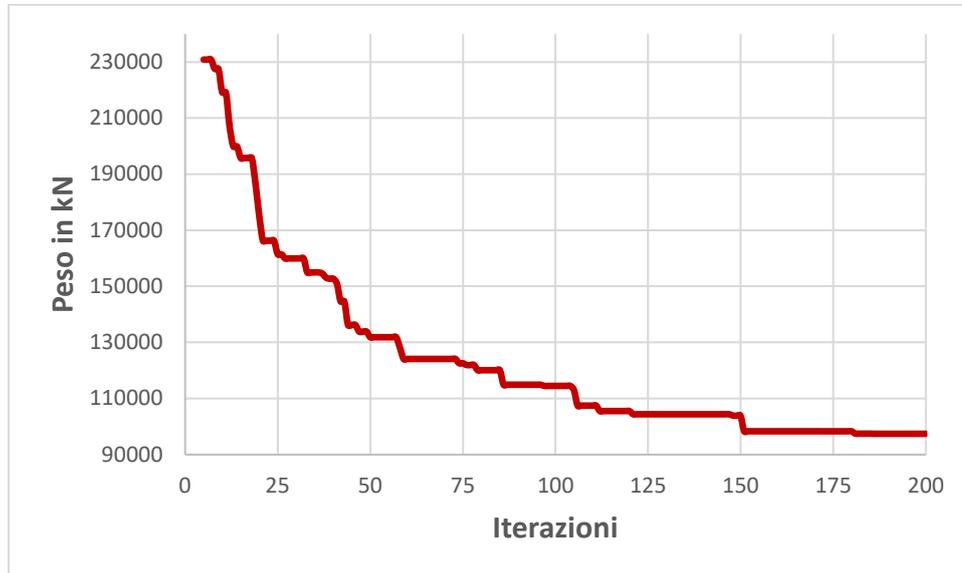


Figura 97 Grafico di andamento della funzione obiettivo.

La soluzione ottimale presenta una massa strutturale pari a 97423 kN. Sono riportate le analisi condotte da Sap2000 per la soluzione ottimale per i vari casi di carico allo stato limite ultimo e allo stato limite di servizio.

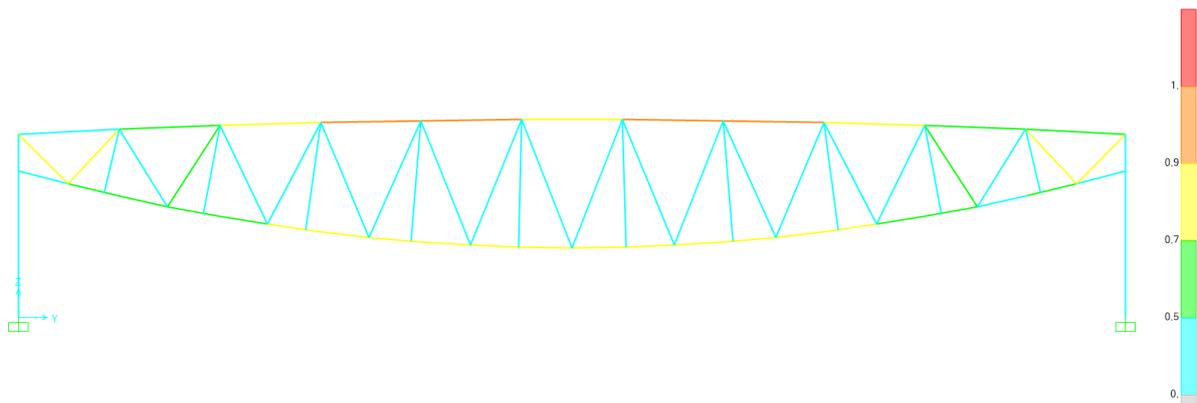


Figura 98 Risultati verifiche strutturali su travatura reticolare vista 2D.

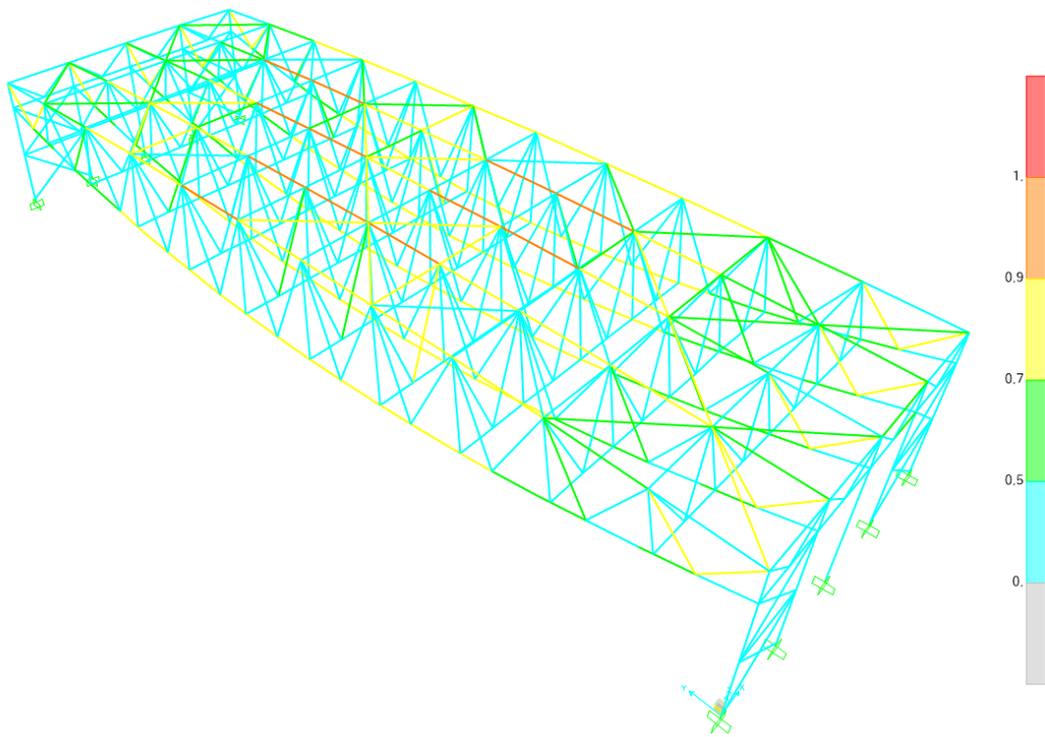


Figura 99 Risultati verifiche strutturali su travatura reticolare vista 3D.

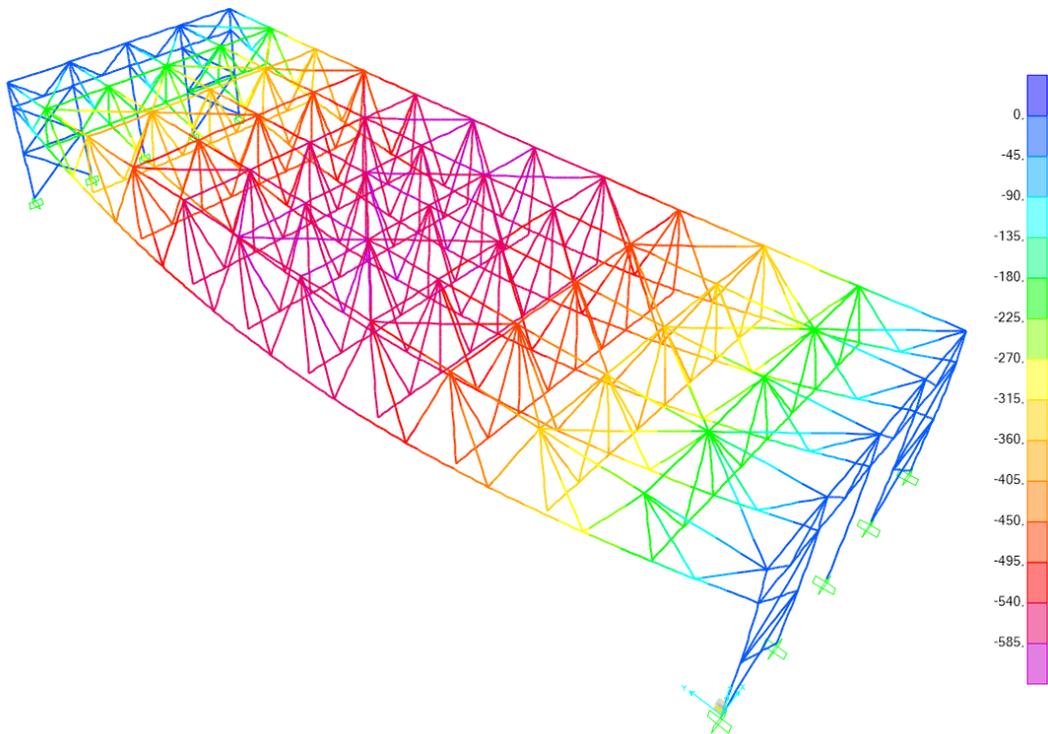


Figura 100 Deformata della travatura reticolare sotto il caso di carico 1.0D+1.0L.

Sono riportate le soluzioni per i vari parametri impostati.

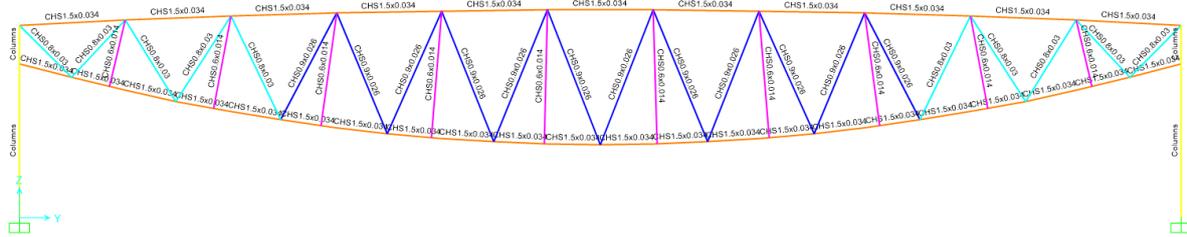


Figura 101 Profili risultanti per la soluzione migliore.

Profili risultanti sono riportati a seconda del gruppo strutturale di appartenenza:

- Diagonali CHS 800x30 mm, CHS 900x26 mm.
- Correnti CHS 1500x34mm, 1500x34mm.
- Montanti CHS 600x14mm.

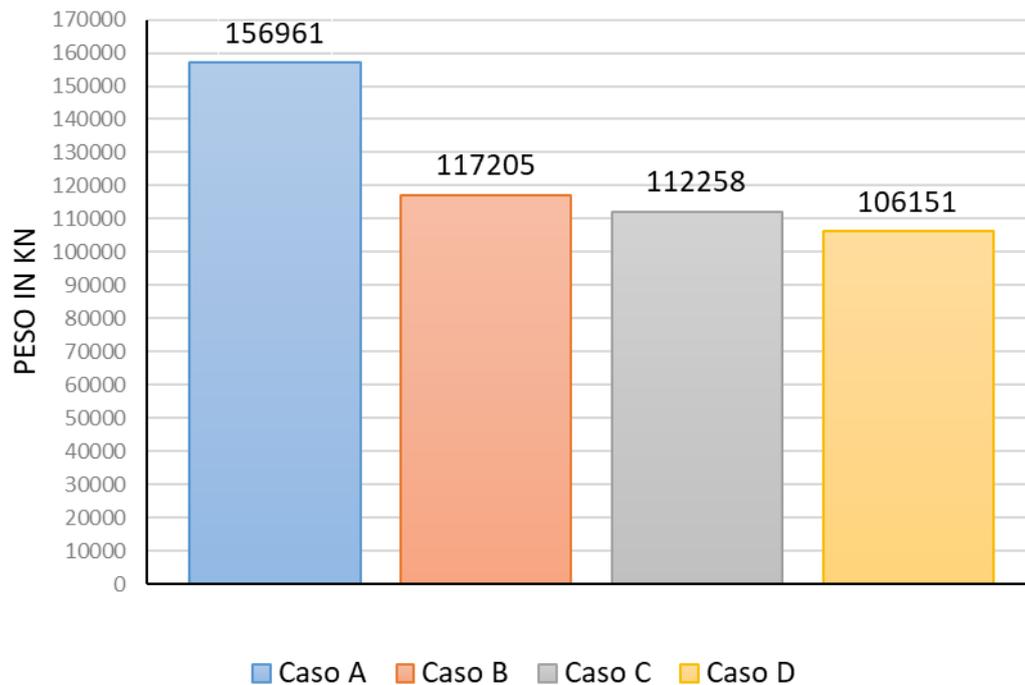
Parametri geometrici risultati:

- Distanza dalla corda 4m corrente superiore.
- Distanza dalla corda -25m corrente superiore.

È risultata una curvatura del corrente superiore pari a  $0.033 \frac{1}{m}$  e una curvatura per il corrente inferiore pari a  $0.058 \frac{1}{m}$ .

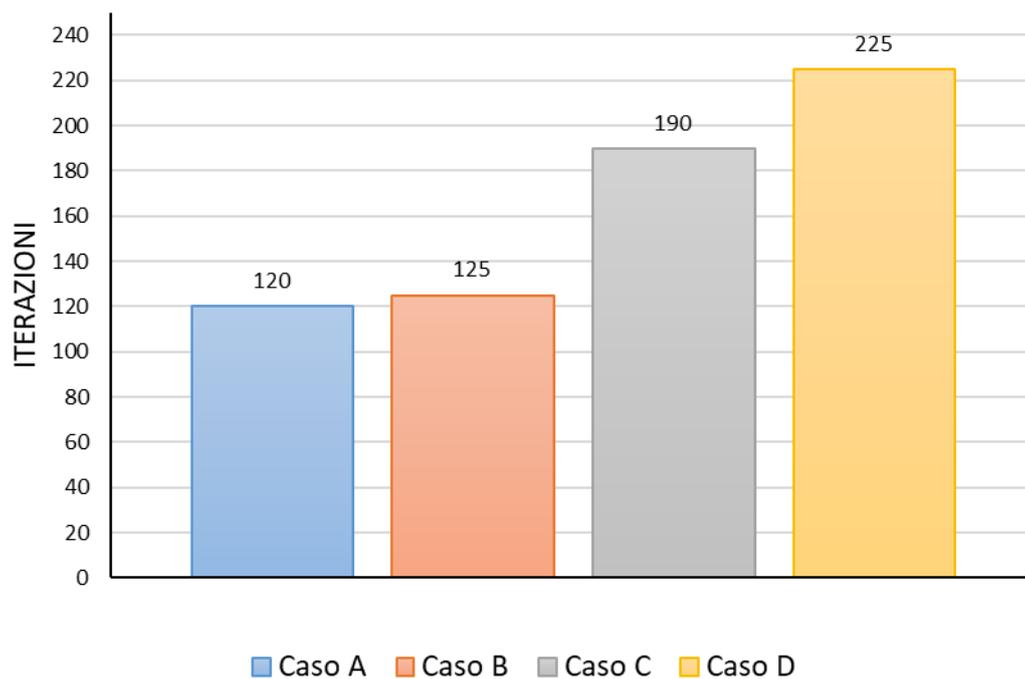
### 5.4.3 RISULTATI E CONFRONTO

I risultati ottenuti dalle analisi del caso studio 1 hanno mostrato un ottimo comportamento dell' algoritmo di ottimizzazione EDA nei confronti dei problemi affrontati mostrando dei valori in linea con quelli ipotizzati.



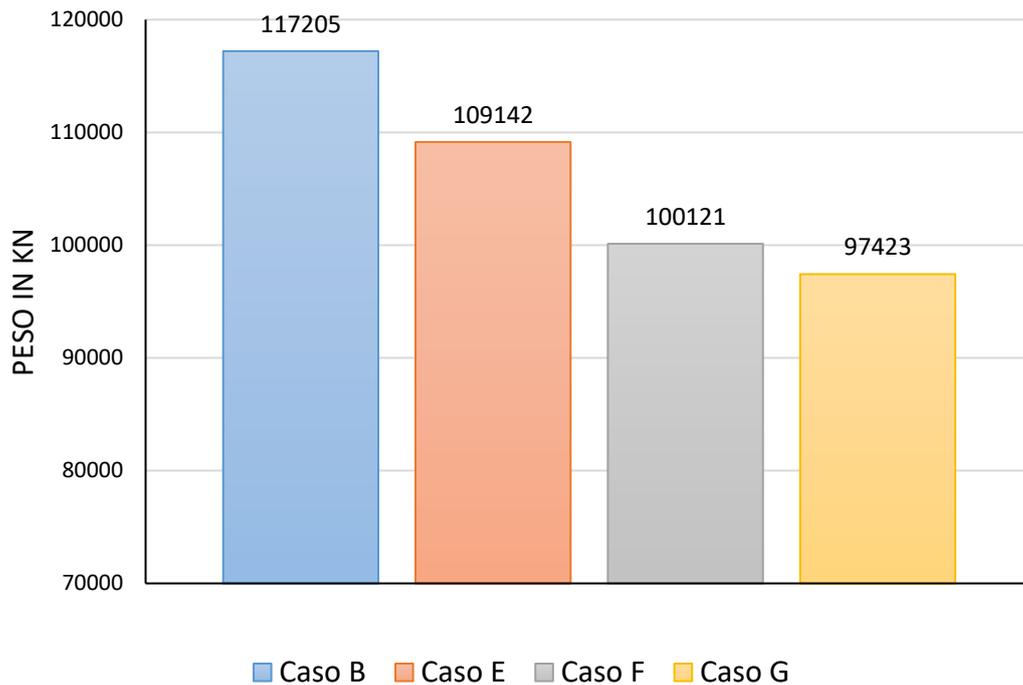
*Peso della struttura per i casi rappresentati*

Come prevedibile un incremento dei parametri meccanici legato alla scelta dei profili, per vari gruppi di aste, porta a una diminuzione del peso complessivo della struttura poiché rende il processo di ottimizzazione più accurato. Il crescere del numero dei parametri ha anche portato sì a soluzioni sempre più ottimizzate ma con uno scarto della variazione della soluzione decrescente. Infatti, vediamo come già la scelta di soli cinque parametri, per la travatura reticolare, porti con buona approssimazione a un'ottimizzazione accettabile sia da un punto di vista computazionale che da un punto di vista esecutivo, richiedendo una bassa scelta dei profili.



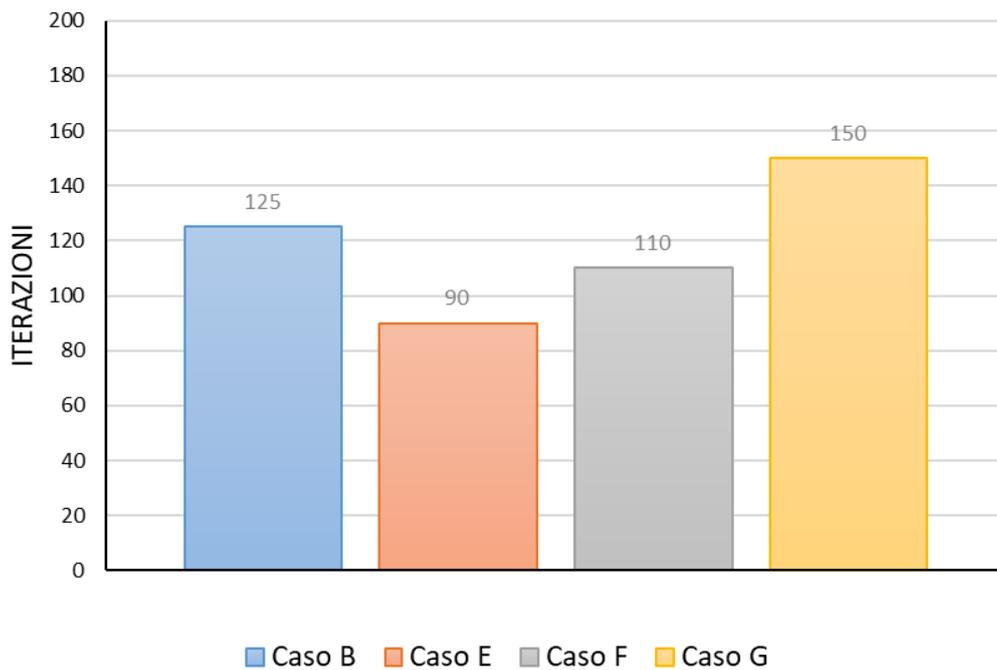
*Numero di iterazione per raggiungere il punto di ottimo*

Dall'analisi dei risultati è evidente anche come sia più arduo per il processo di ottimizzazione raggiungere il punto di ottimo al crescere del numero di parametri; aumentando infatti il numero di possibilità e di configurazioni che può avere la struttura, si ha un aumento del dominio delle soluzioni da esplorare e di conseguenza un incremento dell'onere computazionale.



*Peso della struttura per i casi rappresentati*

I risultati ottenuti dalle analisi del caso studio 2 hanno mostrato un incremento delle prestazioni dell'ottimizzazione dovuto all'introduzione di parametri geometrici che permettono la variazione della forma della struttura. La variazione della geometria ha portato a un'ottimizzazione più accurata della struttura mostrando un evidente miglioramento anche del processo di ottimizzazione. La variazione della curvatura dei correnti, ridistribuendo l'andamento delle forze interne alla reticolare, genera un ampio numero di soluzioni accettabili e ottimali che porta, con un minor numero di parametri, a soluzioni ottimali migliori di una geometria fissa.



*Numero di iterazione per raggiungere il punto di ottimo*

I risultati del caso G mostrano il corretto funzionamento del processo di ottimizzazione andando a generare una forma ottimale per la travatura reticolare in linea con quanto era stato ipotizzato. Infatti, i risultati dell'ottimizzazione propongono, una soluzione con una curvatura minima per il corrente superiore, in modo da minimizzare la superficie di carico della struttura, e un'ampia curvatura per il corrente inferiore in modo da generare, un sufficiente momento resistente ai carichi imposti.



*Figura 102* Trave reticolare stadio Al-Janoub, similitudine con caso G.

Le fasi preliminari di ogni processo di ottimizzazione hanno visto il lavoro congiunto del Latin Hypercube Sampling e della funzione di penalizzazione portando con successo alla generazione automatica di individui “feasible” creando un processo iterativo senza alcun bisogno di dati in entrata. Ciò ha comunque portato ad un onere computazionali maggiore visto che l’introduzione di un individuo “feasible” fin dalle prime iterazioni avrebbe portato a una velocizzazione del processo ma in tal modo siamo riusciti a testare, su una situazione reale, l’efficacia del metodo proposto.

Da un punto di vista dei costi delle soluzioni proposte, considerando un confronto tra la soluzione di ottimo del caso A e la soluzione proposta del caso F, è evidente come l’impatto complessivo sui costi della struttura sia enorme arrivando a un risparmio di complessivo 46.040.400€, ipotizzando un costo dell’acciaio 900 €/ton.

	Peso modulo [ton]	Peso totale [ton]	Costo acciaio [€]
Caso A	15696.1	141264.9	127138410
Caso B	11720.5	105484.5	94936050
Caso C	11225.8	101032.2	90928980
Caso D	10615.1	95535.9	85982310
Caso E	10914.2	98227.8	88405020
Caso F	10012.1	90108.9	81098010
Caso G	9742.3	87680.7	78912630

*Tabella con risultati dei casi studio.*

## 6 CONCLUSIONI

Dal lavoro di tesi è stato sviluppato un algoritmo EDA funzionante in linguaggio di programmazione Matlab e Python che ha permesso la risoluzione del caso di studio preso in esame.

Nei primi capitoli è stata testata l'efficacia dell'algoritmo proposto mostrando buoni risultati a confronto di algoritmi testati e sviluppati da tempo, presenti in commercio e nelle librerie open-sources. La funzione di penalizzazione proposta per la risoluzione dei problemi vincolati ha permesso all'algoritmo di utilizzare a pieno le proprie potenzialità così da non scartare nessuna informazione data dalle generazioni di individui analizzati e localizzando con buona efficacia la frontiera della zona "feasible". Ciò ha permesso di generare individui "unfeasible" che potessero aiutare nella ricerca di ottimo per problemi strutturali, dal momento che le condizioni ottimali di una struttura si trovano in concomitanza dalla frontiera dei vincoli imposti dalle analisi strutturali.

Si sono però riscontrate criticità nell'applicazione dell'EDA in problemi a variabili discontinue dovute mostrando invece un'ottima efficacia in problemi a variabili continue. Questo è dovuto alla natura dell'EDA e al tipo di modello probabilistico sviluppato al suo interno. Ciò però non ha impedito di ricavare ottimi risultati da questo tipo di problemi dal momento che si è rivelato un ottimo strumento, se ben impostato, l'utilizzo della matrice di varianza.

I risultati dei casi studio hanno a pieno validato il corretto funzionamento dell'algoritmo sviluppato mostrando risultati in linea con quelli attesi. Dall'applicazione dell'EDA al caso di studio si è potuto formulare e definire un approccio parametrico al problema creando strumenti di ottimizzazione che permettono di implementare l'ottimizzazione in un ambiente, Grasshopper, ormai familiare agli ingegneri strutturalisti ed edili. Tutto ciò è nato grazie alla collaborazione con lo studio di progettazione Maffei Engineering SpA dalla quale ne è derivata, grazie agli strumenti messi a disposizione, la creazione di un unico modello strutturale per l'analisi del caso studio. Il metodo di lavoro sviluppato tramite il software Grasshopper ha permesso di gestire a pieno ogni singolo parametro, di elaborare i risultati delle analisi, di controllare il corretto funzionamento del processo di ottimizzazione e di creare tool su misura per implementare il processo di ottimizzazione strutturale.

Infine, i risultati delle analisi hanno permesso di scovare soluzioni con forme e configurazioni differenti per la struttura permettendo quindi un'ampia scelta da parte del progettista sulla soluzione da adottare. Un risultato lampante è stato quello di raggiungere già un risultato ottimale utilizzando un relativamente basso numero di parametri. La configurazione parametrica del caso B è una delle tipiche soluzioni ingegneristiche con un limitato numero di profili così che rendano una struttura facile sia da un punto di vista della fabbricazione che per la pianificazione degli ordini dei profili. Per questo motivo ancora più ottimi sono stati i risultati delle parametrizzazioni geometriche. Partendo dalla stessa configurazione del Caso B siamo arrivati a soluzioni ingegneristiche ancora più prestanti e leggere modificando la forma in modo da assorbire in maniera ottimale i carichi imposti alla struttura.

## **6.1 SVILUPPI FUTURI**

Dal lavoro intrapreso con la seguente tesi sono emerse molte peculiarità sulle funzionalità e potenzialità degli EDA soprattutto nella possibilità di generare modelli probabilistici differenti e che possano mutare ed evolversi a seconda del tipo di problema. Uno dei possibili sviluppi scaturiti da questa tesi sarà il potenziamento e in seguito la messa a disposizione di una componente Grasshopper che possa risolvere problemi di ottimizzazione tramite algoritmo EDA.

## 7 BIBLIOGRAFIA

- [1] A. A. J. C. B. O. C. a. W. D. K. Manuel Chica, *hy simheuristics? Benefits, limitations, and best practices when combining metaheuristics with simulation*, 2020.
- [2] M. W. F. G. L. Martin Pelikan, Estimation of Distribution Algorithms..
- [3] M. W. Martin Pelikan, An introduction and Survey of Estimation of Distribution Algorithms..
- [4] R. Iman, J. Davenport e D. Zeigler, Latin hypercube sampling (program user's guide), 1980.
- [5] G. J. & B. K. McLachlan, Mixture models: Inference and applications to clustering(Vol. 38), New York: M Dekker, 1988.
- [6] G. & P. D. McLachlan, Finite mixture models, New York: Wiley, 2000.
- [7] Rosso, Melchiorre, Cucuzza, Manuello e Marano, «Estimation of distribution algorithm for constrained optimization in structural design.».
- [8] X.-S. Y. Momin Jamil, A Literature Survey of Benchmark Functions For Global Optimization Problems.
- [9] L. P. K. a. C. H. ChenNanyang, «Integration of Response Surface Methodology with Genetic Algorithms,» *Technological University, School of Mechanical and Production Engineering, Singapore*, 2001.
- [10] P. S. C. G. Camp C, «Optimized design of two-dimensional structures using genetic algorithm,» *J Struct Eng* , 1998.
- [11] Mahfouz SY, «Design optimization of structural steelwork,» Ph.D. thesis, Department of Civil and Environmental Engineering, University of Bradford, UK, 1999.
- [12] C. C. A. C. Helio J.C.Barbosa, «A genetic algorithm encoding for cardinality constraints and automatic variable linking in structural optimization,» *Engineering Structures*, vol. 38, n. 12, pp. 3708-3723, 2008.
- [13] B. B. Camp CV, «5] Camp CV, Bichon BJ. Design of space trusses using ant colony optimization. 741-51,» *J Struct Eng*, 2004.
- [14] S. M., « Artificial bee colony algorithm for optimization of truss structures.,» *Appl Soft Comput* 2011;11(2):2406-18.
- [15] M. F. C.V. Camp, «Design of space trusses using modified teaching–learning based optimization».
- [16] M. H. Schmit LA, « A new structural analysis/synthesis capability-ACCESS1 AIAAJ,» vol. 14, pp. 661-71, 1974.
- [17] M. Z. Stanislaw, *Strutture spaziali in acciaio*, Milano: Cisia, 1963.
- [18] Numpy, <https://numpy.org/doc/stable/index.html>.
- [19] IronPython, <https://ironpython.net/>.
- [20] M. A. Rahman, *GH\_CPython* [https://github.com/MahmoudAbdelRahman/GH\\_CPython](https://github.com/MahmoudAbdelRahman/GH_CPython).
- [21] A. Distribution, <https://www.anaconda.com/products/distribution>.
- [22] Anemone, <https://www.food4rhino.com/en/app/anemone>.
- [23] Scipy, <https://scipy.org/>.

Figura 1 Diagramma di flusso dell’algoritmo EDA.....	12
Figura 2 A sinistra campionamento dominio con Latin Hypercube, a destra campionamento random.....	14
Figura 3 Generatore di numeri random: popolazione di partenza 100 individui. ....	15
Figura 4 Generatore di numeri random: popolazione di partenza 200 individui. ....	16
Figura 5 Latin hypercube sampling: popolazione di partenza 100 individui. ....	16
Figura 6 Selezione degli individui: disposizione e troncamento del campione .....	18
Figura 7 Esempio 1D su funzione di Rastring: in rosso gli individui scartati, in nero selezionati.....	19
Figura 8 Modello GMM su dominio bidimensionale.....	20
Figura 9 Distribuzione di probabilità iniziale costante. ....	22
Figura 10 Generazione e selezione nuova popolazione: in rosso gli individui scartati, in nero gli individui selezionati. ....	22
Figura 11 Gaussian Mixture Model: generazione del modello dagli individui selezionati .....	23
Figura 12 Gaussian Mixture Model: individui con diversa pesatura $\pi_i$ . ....	24
Figura 13 Variazione dinamica di $\sigma$ in funzione di $\beta$ .....	26
Figura 14 Rappresentazione bidimensionale della funzione sferica. ....	33
Figura 15 In rosso la prima generazione di individui.....	34
Figura 16 In rosso la miglior soluzione dopo la duecentesima iterazione. ....	34
Figura 17 Rappresentazione grafica della funzione di Himmelblau. ....	36
Figura 18 Rappresentazione grafica della funzione di Ackley. ....	38
Figura 19 In rosso la prima generazione di individui.....	39
Figura 20 In rosso la miglior soluzione dopo la duecentesima iterazione. ....	39
Figura 21 Rappresentazione bidimensionale della funzione di Rastringin. ....	41
Figura 22 In rosso la prima generazione di individui.....	42
Figura 23 In rosso l’ottimo della funzione calcolato dall’EDA.....	42
Figura 24 Valutazione funzione penalty su problema di Sickle: in rosso il dominio della funzione obiettivo, in ciano la frontiera dei vincoli imposti dal problema e in nero gli individui “unfeasible” valutati con il metodo di penalizzazione adattivo proposto.....	47
Figura 25 Problema di Sickle: in rosso il dominio della funzione obiettivo, in nero la rappresentazione delle funzioni di vincolo che racchiudono la zona “feasible”.....	49
Figura 26 Problema di Sickle: dettaglio zona “feasible”. ....	49
Figura 27 Geometria e schema statico del problema. ....	52
Figura 28 Tipologie di travature reticolari. ....	57
Figura 29 Copertura dell’acciaieria ILVA a Taranto, luce complessiva 254 metri. ....	57
Figura 30 Caso studio: Dimensioni struttura.....	58
Figura 31 Schematizzazione della struttura .....	58
Figura 32 Dimensioni caso studio .....	59
Figura 33 Vincoli della struttura. ....	60
Figura 34 Carico variabile applicato alla copertura $kN/m^2$ .....	61
Figura 35 FeMM Maffeis interoperabilità .....	63
Figura 36 Ambiente Grasshopper. ....	64
Figura 37 Flusso dei dati per il processo di ottimizzazione e le varie operazioni.....	66
Figura 38 Modello geometrico .....	67
Figura 39 Possibili configurazioni geometriche: variazione della curvatura del corrente superiore e inferiore. ....	68
Figura 40 Possibili configurazioni geometriche: variazione inclinazione dei diagonali. ....	68

Figura 41 Assegnazione delle proprietà meccaniche. ....	70
Figura 42 Analisi strutturale con tool FeMM. ....	70
Figura 43 Risultati verifiche modello FEM in Sap2000 .....	71
Figura 44 Impostazione verifiche normativa AISC 360-10. ....	71
Figura 45 Script per calcolo della funzione obiettivo e analisi dei vincoli strutturali. ....	72
Figura 46 Componente Grasshopper per Latin Hypercube Sampling. In entrata il numero di individui da generare e la matrice contenente il dominio per ogni singolo parametro, in uscita la popolazione iniziale. ....	73
Figura 47 Componente Grasshopper per ottimizzazione tramite algoritmo EDA. ....	74
Figura 48 Flusso di dati dell'ottimizzatore. ....	75
Figura 49 Componente Anemone che permette di creare un processo iterativo. ....	75
Figura 50 Processo iniziale di valutazione di N individui generati tramite Latin Hypercube Sampling. ....	76
Figura 51 Loop del processo di ottimizzazione: il processo prevede una popolazione iniziale valutata come dato in entrata. K equivale al numero massimo di iterazioni impostato per l'analisi e N al numero di individui totale della popolazione. ....	77
Figura 52 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico. ....	80
Figura 53 Grafico di andamento della funzione obiettivo. ....	81
Figura 54 Risultati verifiche strutturali su travatura reticolare vista 2D. ....	81
Figura 55 Risultati verifiche strutturali su travatura reticolare vista 3D. ....	82
Figura 56 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L. ....	82
Figura 57 Profili risultanti per la soluzione migliore. ....	83
Figura 58 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico. ....	84
Figura 59 Grafico di andamento della funzione obiettivo. ....	85
Figura 60 Risultati verifiche strutturali su travatura reticolare vista 2D. ....	85
Figura 61 Risultati verifiche strutturali su travatura reticolare vista 3D. ....	86
Figura 62 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L. ....	86
Figura 63 Profili risultanti per la soluzione migliore. ....	87
Figura 64 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico. ....	88
Figura 65 Grafico di andamento della funzione obiettivo. ....	89
Figura 66 Risultati verifiche strutturali su travatura reticolare vista 2D. ....	89
Figura 67 Risultati verifiche strutturali su travatura reticolare vista 3D. ....	90
Figura 68 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L. ....	90
Figura 69 Profili risultanti per la soluzione migliore. ....	91
Figura 70 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico. ....	92
Figura 71 Grafico di andamento della funzione obiettivo. ....	93
Figura 72 Risultati verifiche strutturali su travatura reticolare vista 2D. ....	93
Figura 73 Risultati verifiche strutturali su travatura reticolare vista 3D. ....	94
Figura 74 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L. ....	94
Figura 75 Profili risultanti per la soluzione migliore. ....	95
Figura 76 Impostazioni dei parametri geometrici per la variazione della curvatura sui correnti. ....	96
Figura 77 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico. ....	97
Figura 78 Parametrizzazione geometrica curvatura corrente superiore. ....	97
Figura 79 Possibili configurazioni geometriche. ....	97
Figura 80 Grafico di andamento della funzione obiettivo. ....	98
Figura 81 Risultati verifiche strutturali su travatura reticolare vista 2D. ....	99
Figura 82 Risultati verifiche strutturali su travatura reticolare vista 3D. ....	99

Figura 83 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L.....	100
Figura 84 Profili risultanti per la soluzione migliore. ....	100
Figura 85 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico. .....	101
Figura 86 Parametrizzazione geometrica curvatura corrente inferiore. ....	101
Figura 87 Possibili configurazioni geometriche.....	102
Figura 88 Grafico di andamento della funzione obiettivo.....	103
Figura 89 Risultati verifiche strutturali su travatura reticolare vista 2D.....	103
Figura 90 Risultati verifiche strutturali su travatura reticolare vista 3D.....	104
Figura 91 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L.....	104
Figura 92 Profili risultanti per la soluzione migliore. ....	105
Figura 93 Gli elementi con lo stesso colore appartengono ognuno a un gruppo parametrico. .....	106
Figura 94 Parametrizzazione geometrica curvatura corrente inferiore. ....	106
Figura 95 Parametrizzazione geometrica curvatura corrente superiore. ....	107
Figura 96 Possibili configurazioni geometriche.....	107
Figura 97 Grafico di andamento della funzione obiettivo.....	108
Figura 98 Risultati verifiche strutturali su travatura reticolare vista 2D.....	108
Figura 99 Risultati verifiche strutturali su travatura reticolare vista 3D.....	109
Figura 100 Deformazioni della travatura reticolare sotto il caso di carico 1.0D+1.0L.....	109
Figura 101 Profili risultanti per la soluzione migliore. ....	110
Figura 102 Trave reticolare stadio Al-Janoub, similitudine con caso G.....	114