

UNIVERSITÀ DEGLI STUDI DI MILANO
Facoltà di Scienze e Tecnologie
Corso di Laurea in Informatica

TEMP IDENTIFICAZIONE DI SATELLITI

Relatore: Prof. Giovanni RIGHINI

Correlatore:

Tesi di:
Jonathan Junior AGYEKUM
Matricola: 935132

Anno Accademico 2022-2023

dedicato a ...

Prefazione

prefazione

Organizzazione della tesi

La tesi è organizzata come segue:

- nel Capitolo 1

Ringraziamenti

Grazie.

Indice

	ii
Prefazione	iii
Ringraziamenti	iv
1 Introduzione	1
1.1 Strumenti utilizzati	1
1.2 Ricerca operativa e astronomia	1
1.3 Storia dell'identificazione di satelliti	1
1.3.1 Rilevamenti diretti e indiretti	1
2 Descrizione del problema	2
2.1 Descrizione	2
2.2 Campi di applicazione e utilità	3
3 Sviluppo del modello	4
3.1 Definizione del modello	4
3.1.1 Calcolo della regione pareto-ottima	5
3.2 Algoritmo di controllo	7
3.2.1 Definizione del passo per il minimo periodo β	7
3.2.2 Criterio arresto	8
3.2.3 Scelta del solutore	8
3.2.4 Gestione degli errori	11
3.2.5 Selezione della soluzione	12

Capitolo 1

Introduzione

1.1 Strumenti utilizzati

1.2 Ricerca operativa e astronomia

1.3 Storia dell'identificazione di satelliti

1.3.1 Rilevamenti diretti e indiretti

Capitolo 2

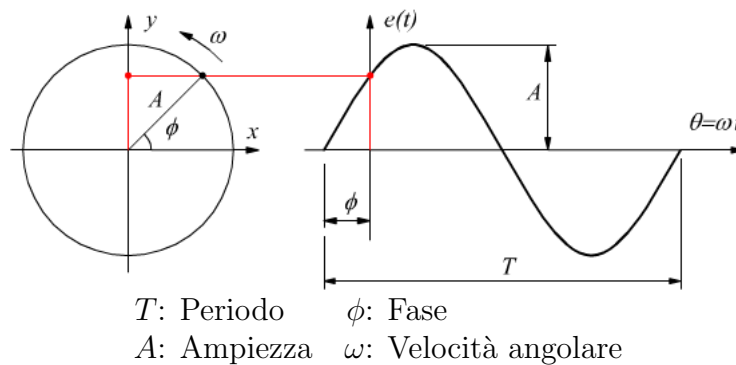
Descrizione del problema

2.1 Descrizione

L'osservazione in modo continuo dell'angolo descritto da k satelliti intorno al loro pianeta, genera delle sinusoidi, questo apre alla possibilità di una identificazione dei satelliti attraverso l'analisi del loro moto orbitale.

Il legame tra l'orbite dei satelliti e le sinusoidi può essere spiegata considerando le orbite dei satelliti circolari e quindi attraverso le leggi del moto circolare.

Figura 1: Moto circolare uniforme



Dove il periodo corrisponde al tempo di rivoluzione del satellite, l'ampiezza alla massima distanza che intercorre nel periodo tra il satellite e il corrispettivo pianeta, $e(t)$ lo spostamento del satellite lungo l'orbita. È quindi possibile descrivere la sinusoide risultante tramite l'espressione analitica: $e(t) = A \sin(\omega t + \phi)$

L'obiettivo è quello di riuscire ad attribuire ai k satelliti i k valori di ogni osservazione e di trovare periodo, ampiezza e fase delle k sinusoidi, supponendo di poter

osservare i k valori simultaneamente, in n momenti successivi. Si suppone di non conoscere nulla dei k satelliti e soprattutto di non poterli distinguere in fase di rilevazione dei valori.

Il problema è scomponibile in due sottoproblemi: uno di interpolazione e uno di assegnamento. La componente di interpolazione è identificata nell'operazione di individuazione della sinusoide che si scosta il meno possibile dai valori. Questo definisce un problema di interpolazione non-lineare e continuo: non-lineare poiché una sinusoide è una funzione periodica non-lineare, definita da una trasformazione elementare della funzione seno, continuo poiché il dominio di una sinusoide e delle sue componenti è definito in \mathbb{R} o in un suo sottoinsieme.

La componente di assegnamento è individuata nel processo di selezione dei punti tra le diverse osservazioni da interpolare. La funzione di questa componente è quella di individuare k sinusoidi, generate dall'interpolazione di una assegnazione di valori, che rappresentino i k satelliti.

2.2 Campi di applicazione e utilità

Il principio della tipologia di identificazione che ho studiato è quello di riconoscere satelliti di un pianeta in base alla variazione in un determinato dominio di una caratteristica quantificabile, in questo caso l'angolo, in funzione del tempo. La variazione dev'essere riconoscibile e identificativa del satellite, cioè deve definire un andamento caratteristico di una funzione elementare o una combinazione di essi. Questo principio è illustrato, per esempio dal metodo di transito, metodo di identificazione fotometrico di satelliti extrasolari [4].

La tipologia di identificazione che illustrerò, può essere applicata sia ai casi di rilevamento diretto, tecniche che permettono di osservare direttamente al telescopio i satelliti, e sia indiretti, cioè l'individuazione tramite effetti fisici che satellite può indurre. Il vincolo fondamentale è che sia possibile effettuare misurazioni dell'angolo rispetto al suo pianeta.

L'identificazione attraverso l'analisi del moto orbitale può essere utile in tutte le situazioni dove non è possibile riconoscere o distinguere due o più satelliti o qualunque entità il quale movimento è osservabile e rotante.

Capitolo 3

Sviluppo del modello

3.1 Definizione del modello

Propongo un modello per la risoluzione del sottoproblema di interpolazione. La sinusoide che si vuole ottenere dall'interpolazione dei punti deve avere determinate caratteristiche. Tra tutte le sinusoidi possibili che interpolano gli n valori, che in questo contesto chiamerò punti, si vuole quella di massimo periodo e sicché le misurazioni possono essere affette da errore si deve combinare la ricerca del massimo periodo con quella di minimo errore, definendo un problema a due obiettivi.

... Motivazioni sulla ricerca della sinusoide con massimo periodo ...

... Attenzione a sinusoidi con basso periodo rispetto alla sinusoide da individuare ...

Il processo risolutivo per un problema a molti obiettivi in conflitto tra loro, cioè che il miglioramento di una comporti un peggioramento di un'altra, prevede:

1. Il calcolo della regione pareto-ottima, cioè l'insieme di soluzioni ammissibili non dominate, denominate anche soluzioni paretiane.
2. La scelta di una soluzione tra quelle individuate.

La determinazione della regione pareto-ottima può essere effettuata in diversi modi. I modelli che ho definito inseguito utilizzano rispettivamente metodo dei pesi e metodo dei vincoli, i due modelli si differenziano solo per vincoli e funzione obiettivo.

Parametri I dati a disposizione del modello sono gli n punti assegnati da interpolare, il quale numero è determinato dal modulo di assegnamento. I punti sono individuati in un piano cartesiano dove l'asse delle ascisse rappresenta il tempo e l'asse delle ordinate l'angolo. I parametri del modello sono quindi le n ascisse t_i e le n ordinate e_i degli n punti, con i da 1 a n .

Per una maggiore semplicità numerica considero i secondi per il tempo mentre rappresento gli angoli in radianti.

Variabili Le variabili sono individuate nelle componenti che determinano una sinusoide, data l'espressione analitica generale di una funzione sinusoidale:

$$e_i = A \sin(\omega t_i + \phi) \quad (1)$$

Dall'espressione ho definito le variabili che rappresentano fase ϕ , ampiezza A e la pulsazione ω .

Inoltre c'è la necessità di rappresentare l'errore che può essere generato dalle misurazioni o da un assegnamento di punti errato. L'espressione analitica diventa:

$$e_i = A \sin(\omega t_i + \phi) + \varepsilon_i \quad (2)$$

Perciò ho definito n variabili aggiuntive che rappresentano l'errore per ogni punto.

3.1.1 Calcolo della regione pareto-ottima

Metodo dei pesi

Il metodo dei pesi consiste nel dare un peso a ciascuna delle funzioni obiettivo, così da ridurre il problema a molti obiettivi in uno di programmazione matematica parametrica.

Vincoli Ho definito come unico vincolo l'equazione (2)

Funzione obiettivo Le due funzioni obiettivo da ottimizzare sono:

$$\max f_1 = \frac{2\pi}{\omega} \quad (3)$$

La massimizzazione del periodo

$$\min f_2 = \sum_{i=1}^n \varepsilon_i^2 \quad (4)$$

La minimizzazione dell'errore complessivo, calcolo l'errore quadratico complessivo per l'eliminazione dei possibili valori negativi che si possono generare.

I pesi definiti per le funzioni f_1 e f_2 sono rispettivamente 1 e -1, quindi ottenendo la funzione obiettivo:

$$\max f = f_1 - f_2 \quad (5)$$

Questa configurazione dei pesi di partenza permette di penalizzare tutte quelle soluzioni ammissibili che massimizzano il periodo ma hanno un errore elevato.

Il processo di individuazione delle soluzioni paretiane, con questo metodo e numero di funzioni, consiste nel eseguire un'analisi parametrica sui pesi, cioè analizzare e individuare soluzioni non dominate al variare dei pesi entro un certo range. In questo caso può essere sufficiente solo effettuare la variazione del peso dell'errore.

Un modello che dà priorità al periodo ottenendo un alto errore è per certo non utilizzabile, perché l'alto errore denota una completa mancanza di relazione tra i punti e la sinusoide individuata, questo spiega l'analisi parametrica solo sul peso dell'errore. È anche possibile non effettuare alcuna variazione dei pesi, ed accontentarsi della prima soluzione individuata tramite la configurazione dei pesi di partenza.

Metodo dei vincoli

Il metodo dei vincoli consiste nell'ottimizzare una delle funzioni obiettivo, trasformando le rimanenti in vincoli, utilizzando un termine noto parametrico.

Vincoli Si ha l'equazione (2) e la trasformazione in vincolo della funzione obiettivo legato al periodo (3)

$$\frac{2\pi}{\omega} \geq \beta \quad (6)$$

Il termine noto β sta ad indicare il minimo periodo ammissibile per una soluzione, questo permette di escludere le soluzioni a basso periodo rendendo inammissibili soluzioni con periodo più piccolo del termine noto.

Funzione obiettivo La funzione obiettivo definita è la funzione (4).

L'individuazione delle soluzioni paretiane è effettuata variando il termine noto parametrico nel vincolo (6). Ho scelto di definire il vincolo (6) perché permette di avere un controllo migliore sull'esplorazione dei periodi possibili della sinusoide da individuare, data dalla possibilità di definire come punto di partenza del modello, un'assegnazione della variabile che rappresenta la pulsazione a partire dal termine noto β .

$$\omega = \frac{2\pi}{\beta} \quad (7)$$

Da un'analisi ipotetica dei range di periodi, che possono essere rilevati nel campo di applicazione in questione, è possibile esplorare e individuare sinusoidi entro il range ipotizzato. Al contrario definendo un vincolo a partire dalla funzione obiettivo dell'errore (4), sarebbe di difficile gestione e modellazione la determinazione della variazione dell'errore, per tutte le tipologie di assegnamenti di punti della quale può essere richiesta l'interpolazione ad esempio: punti completamente non relazionati tra di essi, punti rilevati con un alto errore nel processo di misurazione.

3.2 Algoritmo di controllo

Il modello che ho scelto di implementare è il modello che utilizza il metodo dei vincoli per i vantaggi citati nella sua specificazione. Una volta definito l'implementazione del modello è necessario specificare un algoritmo di controllo e di esecuzione di esso. I compiti sono la definizione del valore del minimo periodo β , nel vincolo (6) ad ogni iterazione e la gestione di eventuali errori.

Algoritmo 1: Algoritmo di controllo del modello

```

input : pt Collezione di punti da interpolare  $[(t_1, e_1), \dots, (t_k, e_k)]$ 
output: soluzioni Insieme di soluzioni che interpolano i punti pt

1 passo  $\leftarrow 1$ 
2  $\beta \leftarrow periodoIniziale$ 
3 soluzioni  $\leftarrow \emptyset$ 
4 while  $\beta \leq periodoLimite$  do
5   solPrecedente  $\leftarrow sol$ 
6   sol  $\leftarrow Interpolare(pt, \beta, solutore)$ 
7   Aggiungi sol a soluzioni
8   if sol[periodo] = solPrecedente[periodo] then
9     | passo = f(passo)
10  else
11    | passo  $\leftarrow 1$ 
12  end
13   $\beta \leftarrow \beta + passo$ 
14 end

```

3.2.1 Definizione del passo per il minimo periodo β

Il passo definisce la quantità aggiunta al minimo periodo ad ogni iterazione. Per facilitare l'individuazione di soluzioni differenti, è possibile definire un passo variabile, così da spostare il punto di inizializzazione del problema dalla regione di un minimo locale, in base alla soluzione ottenuta.

L'algoritmo va ad aumentare il passo ogniqualvolta il periodo della soluzione individuata in una iterazione, appartiene ad un intorno del periodo della soluzione dell'iterazione precedente (riga 8), altrimenti si ha un ripristino del passo ad un valore di default (riga 11). In base alle necessità è possibile definire velocità differenti di variazione del passo, tramite la definizione della funzione f (riga 9). Per aumentare il numero di soluzioni individuate è necessario diminuire la velocità di variazione di

passo in modo da ottenere inizialmente piccoli spostamenti del minimo periodo, al contrario per ottenere meno soluzioni ed una esplorazione più veloce, è necessario aumentare la velocità di variazione del passo. esempi possono essere:

$$f(x) = x * k \quad (8)$$

Si moltiplica il passo p per una costante k , è possibile variare la velocità della variazione del passo aumentando o diminuendo il parametro k . Si misura la velocità della funzione di variazione del passo tramite la derivata della funzione $f(x)$.

Un altro fattore che determina l'esplorazione è la dimensione dell'intorno. È necessario definire un intorno poiché è possibile che con approssimazioni e specifiche interne del solutore o anche in base alle caratteristiche della regione ammissibile, non si individui l'esatto valore di un minimo locale, ma un valore nell'intorno di esso in iterazioni successive. Perciò è necessario definire un livello di tolleranza con il quale definire se due soluzioni sono da considerare uguali (riga 8).

3.2.2 Criterio arresto

È necessario specificare un criterio di arresto per l'algoritmo di controllo. L'idea applicata è quella di determinare l'arresto fissando un periodo limite per il minimo periodo β (riga 4). Effettuando una ricerca sui satelliti del sistema solare, ho individuato che il massimo periodo di un satellite nel sistema solare è pari a 9374.0 giorni [3].

Quindi è possibile fissare un periodo limite, effettuando un mappaggio da giorni in secondi, pari a 5000s. Non è necessario fissare esattamente il valore estratto dallo studio dei periodi del sistema solare, poiché il periodo β , utilizzato nel vincolo (6) come limite inferiore, non impedisce al solutore di individuare periodi dai valori molto più grandi rispetto ai periodi del sistema solare, tutto questo facendo partire l'individuazione delle soluzioni da un periodo pari a 0, escludendo i periodi negativi. L'esclusione dei periodi negativi non influisce sull'individuazione delle soluzioni poiché i punti sono rilevati all'avanzare del tempo.

3.2.3 Scelta del solutore

A causa dei vincoli di utilizzo dei solutori globali in `nlopt` [1], la scelta del solutore è ricaduta ai solutori locali (riga 6). È possibile condurre la selezione dell'algoritmo solutore valutandoli su problemi di interpolazione definiti da punti appartenenti ad una stessa sinusoide $e(t) = \sin(\omega t)$ per un numero di punti definito. Le ascisse dei n punti assumono i valori interi dell'insieme $[0, n]$. Analizzo il numero totale di soluzioni individuate, il numero di soluzioni individuate in un intorno del periodo della sinusoide del problema, infine il tempo di calcolo. L'intorno scelto è di dimensioni

pari a 100s, è possibile scegliere un diverso intorno in base alle proprie necessità di precisione rispetto alla sinusoide da individuare.

I risultati delle sperimentazioni sono illustrati nelle tabelle da 1 a 5. I parametri dell'algoritmo 1 per le seguenti sperimentazioni sono:

- $f(x)$ in riga 9, algoritmo 1 pari alla funzione (8) con $k = 2$
- $periodoIniziale = 0s$
- $periodoFinale = 5000s$

Tabella 1: Prestazioni dei solutori: Sinusoide con $\omega = 0.8 \text{ rad/s}$

Solutore	n. soluzioni individuate	n. soluzioni nell'intorno	Tempo di calcolo (s)
COBYLA	120	21	6.1
BOBYQA	19	6	0.9
NEWUOA	113	101	5.9
PRAXIS	139	139	6.9
SBPLX	26	9	1.3

Tabella 2: Prestazioni dei solutori: Sinusoide con $\omega = 0.0125 \text{ rad/s}$

Solutore	n. soluzioni individuate	n. soluzioni nell'intorno	Tempo di calcolo (s)
COBYLA	141	19	7.1
BOBYQA	20	0	1.0
NEWUOA	114	0	5.7
PRAXIS	139	0	6.9
SBPLX	26	1	1.3

H

Tabella 4: Prestazioni dei solutori: Sinusoide con $\omega = 0.0013 \text{ rad/s}$

Solutore	n. soluzioni individuate	n. soluzioni nell'intorno	Tempo di calcolo (s)
COBYLA	193	4	9.8
BOBYQA	19	0	0.9
NEWUOA	113	0	5.6
PRAXIS	146	0	7.3
SBPLX	24	0	1.2

Tabella 3: Prestazioni dei solutori: Sinusoide con $\omega = 0.005 \text{ rad/s}$

Solutore	n. soluzioni individuate	n. soluzioni nell'intorno	Tempo di calcolo (s)
COBYLA	218	3	11.0
BOBYQA	20	0	1.0
NEWUOA	113	0	5.6
PRAXIS	149	0	7.4
SBPLX	25	0	1.2

Tabella 5: Prestazioni dei solutori: Sinusoide con $\omega = 0.0010 \text{ rad/s}$

Solutore	n. soluzioni individuate	n. soluzioni nell'intorno	Tempo di calcolo (s)
COBYLA	54	0	2.7
BOBYQA	19	0	0.9
NEWUOA	113	0	5.6
PRAXIS	144	0	7.2
SBPLX	27	0	1.3

L'algoritmo scelto dall'insieme disponibile è COBYLA [2], da come si può notare, è l'unico solutore che individua soluzioni nell'intorno specificato. È da evidenziare la difficoltà per il solutore COBYLA nel individuare soluzioni al di fuori del range di esplorazione fissato per il criterio d'arresto, come da esempio in tabella 5 dove i punti

del problema sono generati da una sinusoide con periodo pari a $T = \frac{2\pi}{\omega} = \frac{2\pi}{0.0010} = 6280s$.

3.2.4 Gestione degli errori

L'unico errore che può verificarsi da tenere in considerazione è `nlopt.RoundoffLimited`. È un evento che specifica una situazione di errore inerente progressivo, cioè l'errore che si commette rappresentando un numero reale con un numero finito di cifre, intrinseco nei calcolatori. Si può verificare quando si esegue l'implementazione del modello per l'interpolazione dei punti in riga 6 dell'algoritmo 1.

Quando si verifica un eccezione di errore inerente, è possibile rieseguire il solutore aumentando i valori di tolleranza sulla precisione della variazione delle variabili e della funzione obiettivo (riga 18, algoritmo 2).

La tolleranza determina il criterio di arresto nel processo di individuazione di un ottimo locale [1], quando la variazione del valore della funzione obiettivo o delle variabili in ogni direzione è minore della tolleranza impostata, il solutore restituirà l'ultima soluzione individuata, perciò aumentando i valori di tolleranza si impedisce la generazione di un errore inerente progressivo, non considerando a priori, valori dalla precisione elevata.

Algoritmo 2: Algoritmo di controllo del modello con gestione degli errori

input : pt Collezione di punti da interpolare $[(t_1, e_1), \dots, (t_k, e_k)]$
output: soluzioni Insieme di soluzioni che interpolano i punti pt

```

1 passo  $\leftarrow 1$ 
2  $\beta \leftarrow valoreIniziale$ 
3  $tolFun \leftarrow tolleranzaDefault$ 
4  $tolVar \leftarrow tolleranzaDefault$ 
5 soluzioni  $\leftarrow \emptyset$ 
6 while  $\beta \leq periodoLimite$  do
7   try:
8     solPrecedente  $\leftarrow sol$ 
9     sol  $\leftarrow Interpola(pt, \beta, solutore, tolFun, tolVar)$ 
10    Aggiungi sol a soluzioni
11    if sol[periodo] = solPrecedente[periodo] then
12      | passo = f(passo)
13    else
14      | passo  $\leftarrow 1$ 
15    end
16     $\beta \leftarrow \beta + passo$ 
17  catch RoundoffLimited:
18    | Aumenta tolFun e tolVar
19  end
20 end

```

3.2.5 Selezione della soluzione

La selezione di una soluzione tra le tante individuate determina un passo fondamentale per tutto il modulo del sottoproblema di interpolazione, poiché si potrebbe incorrere nel rischio di scartare la sinusoide che rappresenta esattamente uno dei k satelliti. I due metodi identificati per questo compito si basano su diverse ipotesi sulla sinusoide ideale e sulle caratteristiche di tutte le sinusoidi individuate, questi sono il criterio del punto di utopia, e criterio degli standard solo sull'errore.

Criterio del punto di utopia

Il punto di utopia è la soluzione che nello spazio degli obiettivi ha come coordinate i valori ottimi di ciascuno. Il problema principale che sorge nello applicare questo

criterio al problema di interpolazione, è che non si conosce il valore ottimo ideale dell'obiettivo rispetto al periodo, mentre per l'errore, il valore è pari a 0.

Questo succede poiché non esiste un limite superiore al periodo che specifichi l'insieme di sinusoidi che violino uno dei vincoli del modello, essi avranno unicamente come conseguenza, se non relazionati ai punti, un alto errore.

L'idea che quindi ho elaborato è quella di determinare il punto di utopia in base ai valori dei periodi delle sinusoidi individuate. La coordinata rispetto all'errore viene fissata a 0, mentre per la coordinata rispettiva al periodo, calcolo la media dei periodi delle sinusoidi. Questo si basa sull'ipotesi che se la maggior parte dei periodi, individuati dal solutore, cade in un intorno di un determinato valore, allora tale valore è il periodo della sinusoide ricercata. Determino la soluzione da selezionare scegliendo la soluzione più vicina, valutando la distanza euclidea di ogni soluzione rispetto al punto di utopia. Illustro esempi di generazione e scelta di una soluzione:

I problemi scelti per la valutazione e analisi del comportamento del criterio del punto di utopia, sono problemi definiti per 10, 20, 30 punti generati da una sinusoide con un determinato periodo. I parametri dell'algoritmo 2 per le seguenti sperimentazioni sono:

- $f(x)$ in riga 9, algoritmo 1 pari alla funzione (8) con $k = 2$
- $periodoIniziale = 0s$
- $periodoFinale = 5000s$
- $tolleranzaDefault = 1e-6$
- $e(t) = \sin(0.0010t)$ sinusoide con periodo pari a: $T = \frac{2\pi}{\omega} = \frac{2\pi}{0.0010} = 6280s$

Tabella 6: periodo da individuare uguale a 6280s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	1045.8	4.6
20	386.8	5.8
30	4158.2	5.5

- $e(t) = \sin(0.0013t)$ sinusoide con periodo pari a: $T = 4830.7s$

Tabella 7: periodo da individuare uguale a 4830.7s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	919.1	5.7
20	321.6	5.0
30	4728.5	10.3

- $e(t) = \sin(0.005t)$ senoide con periodo pari a: $T = 1256s$

Tabella 8: periodo da individuare uguale a 1256s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	922	9.6
20	3679.1	20.9
30	1881.6	11.8

- $e(t) = \sin(0.0125t)$ senoide con periodo pari a: $T = 502.4s$

Tabella 9: periodo da individuare uguale a 502.4s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	521.7	6.8
20	653.4	15.6
30	642.7	7.3

- $e(t) = \sin(0.8t)$ senoide con periodo pari a: $T = 7.85s$

Tabella 10: periodo da individuare uguale a 7.85s






Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	7.8	2.3
20	8909.3	87.0
30	1286.5	6.1

L'esecuzione dell'algoritmo è caratterizzato da uno scostamento medio dal periodo da individuare di $2377s$ e una mediana di $1278s$ ed infine un tempo medio di esecuzione di $13.62s$ e una mediana di $6.8s$.

La mediana e la media dello scostamento denotano un'alta differenza tra la soluzione scelta dall'algoritmo e la sinusoide del problema, la causa di ciò è da individuarsi nel posizionamento del punto di utopia ed una mancata normalizzazione dei domini dell'errore e periodo.

Prendendo ad esempio la tabella 8, in particolare la soluzione individuata per il problema da 20 punti ed il suo grafico delle soluzioni generate, figura 2 e figura 3, si può notare che una soluzione con scostamento minore è stata individuata, ma non scelta dall'algoritmo di selezione poiché la distanza euclidea dal punto di utopia risulta maggiore rispetto alla soluzione scelta.

Legenda per i grafici da figura 2 a figura 8:

-  : soluzione individuata.
-  : soluzione individuata scelta dal criterio.
-  : punto di utopia.
-  : soluzione individuata ideale.
-  : punto della sinusoide del problema.

Da come si può notare dal primo grafico di figura 2 il punto di utopia risulta essere traslato al di fuori della zona dove la maggior parte delle soluzioni sono situate, a causa di soluzioni outlier.

È necessario quindi implementare un metodo di determinazione del periodo del punto di utopia migliore o passare ad un altro criterio di selezione.

Un miglioramento può essere l'implementazione di una media pesata, dove si determina il peso in base all'errore della soluzione, questo permette di dare più rilevanza

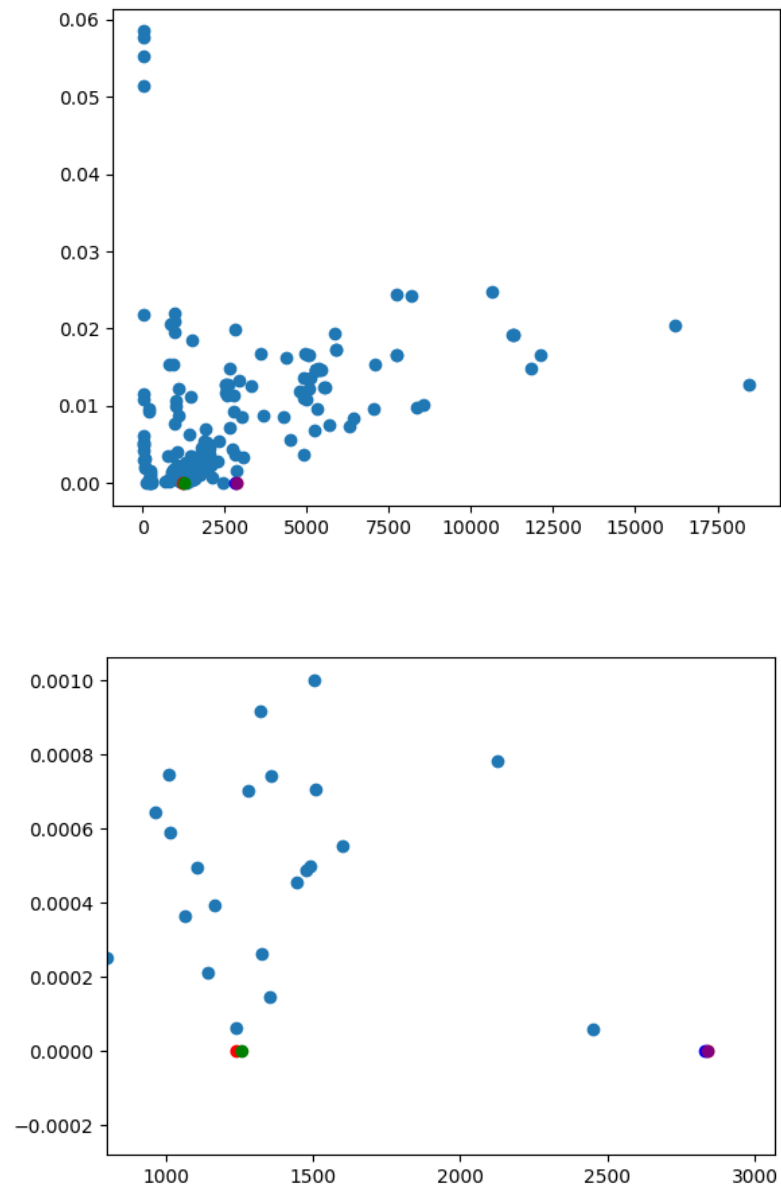


Figura 2: Soluzioni generate per il problema di 8 con 20 punti

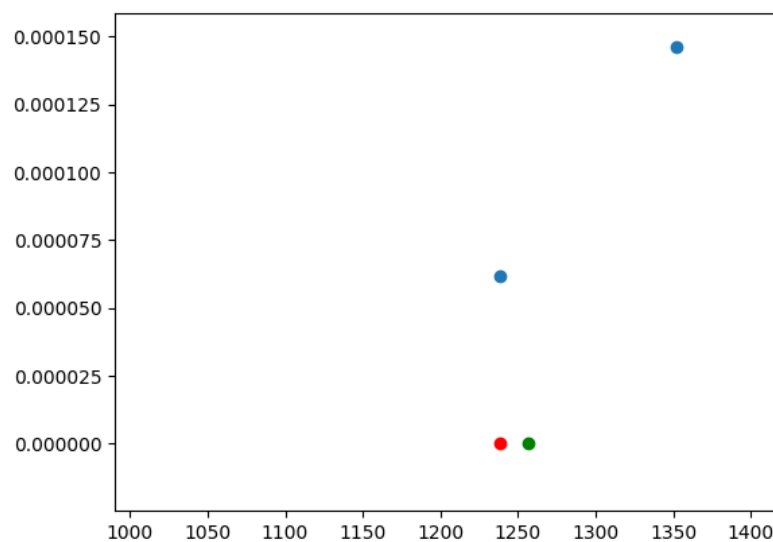


Figura 3: soluzione ideale per il problema di tabella 8 con 20 punti

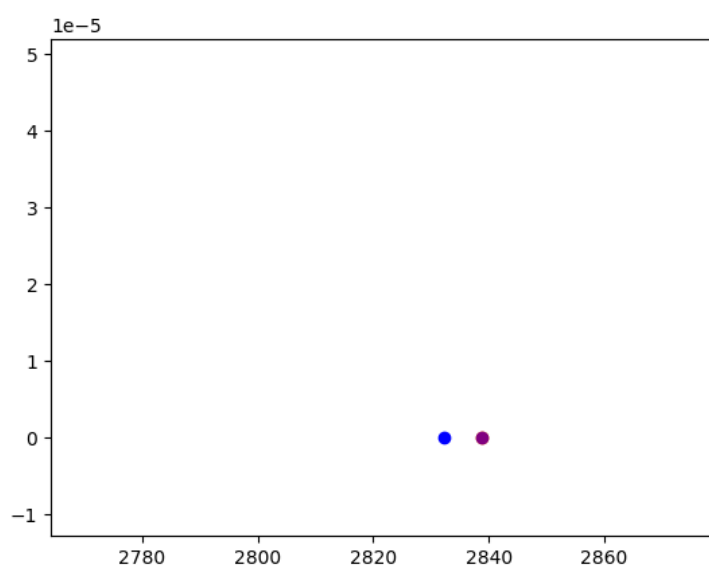


Figura 4: soluzione scelta per il problema di tabella 8 con 20 punti

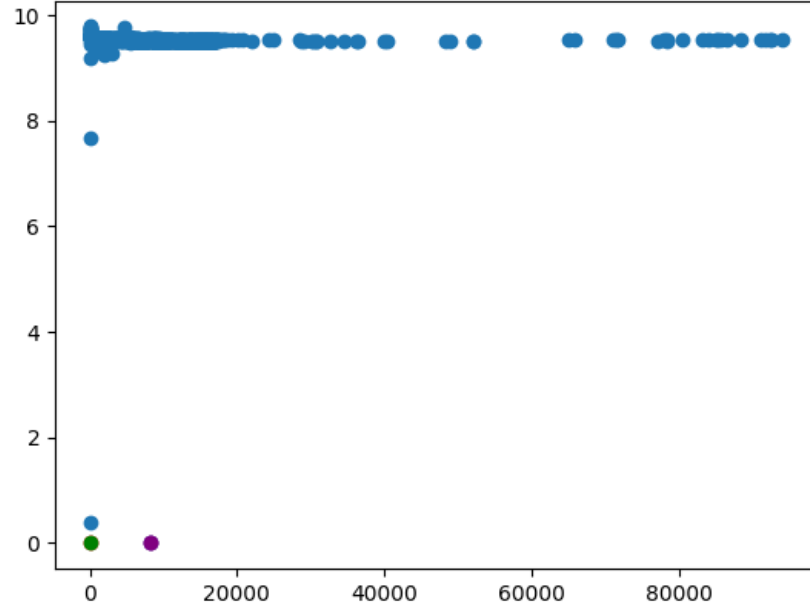


Figura 5: Soluzioni generate per il problema di tabe10 con 20 punti

alle soluzioni con errore minore, comprendendo però in questo modo anche soluzioni con basso periodo rispetto al periodo della sinusoide da individuare.

Una soluzione alternativa può essere la modifica dell'algoritmo di controllo ed esplorazione dei periodi, modificando la gestione del passo. Diminuendo il passo quando si individua una soluzione con errore minore di un determinato valore di soglia, si approfondisce e si aumenta il numero di soluzioni individuate in un intorno di esso.

L'altra probabile causa è la mancata normalizzazione dei domini dell'errore e del periodo. Si possono notare gli effetti di ciò nel grafico in figura 5 corrispondente alla tabella 10 per il problema da 20 punti.

Si può notare, dai grafici in figura 6 e 7 che la distanza tra il punto di utopia e la soluzione con periodo che si scosta meno dalla soluzione da individuare, è approssimativamente pari ad un valore di 8000s, molto maggiore rispetto ad una soluzione che ha lo stesso periodo del punto di utopia ma con uno scostamento maggiore, pari a 9.5

Un altro fattore che ha determinato la mancata scelta della soluzione ideale, mostrata in figura 6, sono il numero di soluzioni individuate in un suo intorno limitato, come per esempio nell'insieme dei periodi $[0, 20]$ illustrato in figura 8, dove sono individuate solo 6 soluzioni rispetto al numero totale di soluzioni individuate pari a 1527

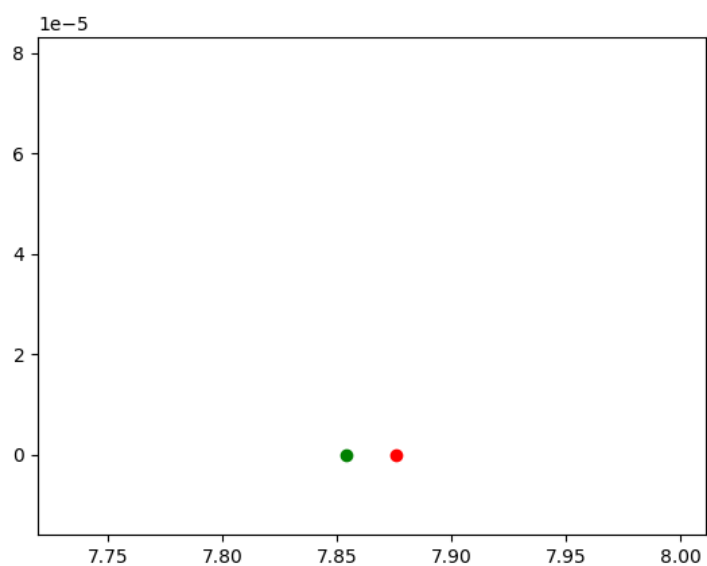


Figura 6: Soluzione ideale per il problema di 10 con 20 punti

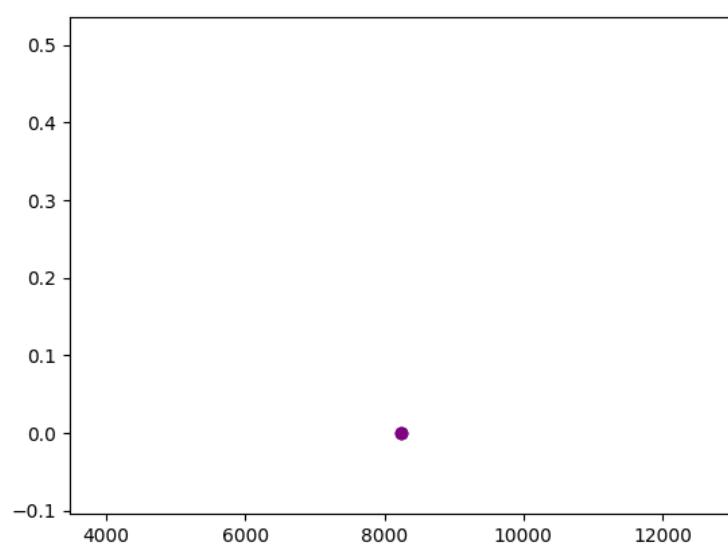


Figura 7: Posizione del punto di utopia per il problema di 10 con 20 punti

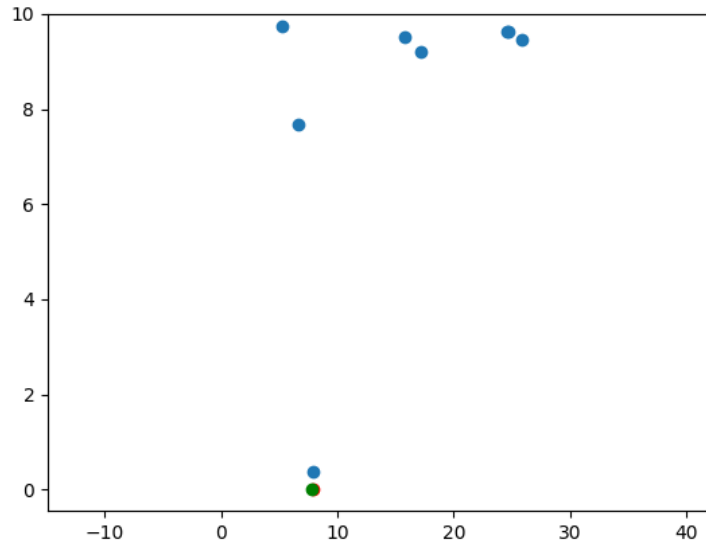


Figura 8: Soluzioni individuate nell'insieme di periodi $[0, 20]$ per il problema di tabella 10 con 20 punti

in questo esempio.

Il problema di questa metodologia di specifica del periodo per il punto di utopia è che è basata in maniera marcata sulle prestazioni, il numero e la tipologia di soluzioni individuate dall'algoritmo di esplorazione dei periodi.

Riassumendo, per il miglioramento del criterio di selezione esplicito le seguenti possibili modifiche:

- La normalizzazione dei domini di errore e di periodo per tutte le soluzioni individuate
- L'implementazione di una media pesata e quindi il calcolo del peso per ogni soluzione individuata in base al suo errore.
- Aggiunta di condizione di diminuzione del passo nel algoritmo esplorazione dei periodi.

Dato le ampie modifiche richieste, ho optato per l'individuazione e definizione di un altro criterio di selezione.

Criterio degli standard sull'errore

Gli standard sono valori-soglia al di sotto dei quali non si vuole che gli obiettivi possano peggiorare. Data la non predicibilità del periodo ottimale in questo problema, è possibile definire gli standard solo per l'errore. La modalità con il quale ho definito gli standard per la scelta della soluzione prevede la suddivisione in range del dominio dell'errore, ipotizzando che ogni soluzione in uno stesso range sia equivalente, se si considera solo l'errore. Ogni range è identificato con un numero ordinale specificandoli come livelli: primo livello, secondo livello, ... , n-esimo livello, dove per ordine crescente dei livelli si ha l'aumento del range d'errore. I range devono essere definiti a priori, in maniera attenta rispetto all'errore intrinseco dello strumento di rilevazione dei punti. È necessario inoltre effettuare ipotesi sul errore di una soluzione: errore di una soluzione ottimale, errore di una soluzione non correlata ai punti, errore di una soluzione legata ad un sottoproblema con un assegnamento di punti errato.

Il seguente pseudocodice illustra una procedura di una struttura dati che ho definito, per contenere la soluzione con livello più basso possibile e con più alto periodo tra tutte le soluzioni individuate all'interno del corrispettivo livello, perciò gli standard dell'errore sono fissati dalla soluzione con il livello minore.

Algoritmo 3: Criterio degli standard

```

input : solCorrente tupla contenente (errore, periodo, ampiezza, fase)
1 if LivelloDi(solCorrente) < LivelloDi(solOttimale) then
2   | solOttimale ← solCorrente
3 else
4   | if LivelloDi(solCorrente) = LivelloDi(solOttimale) then
5     |   if LivelloDi(solCorrente) = ultimoLv then
6       |     if solCorrente[errore] < solOttimale[errore] then
7         |       | solOttimale ← solCorrente
8       |     end
9     |   else
10      |     if solCorrente[periodo] < solOttimale[periodo] then
11        |       | solOttimale ← solCorrente
12        |     end
13      |   end
14    end
15 end

```

Tutte le soluzioni con un livello maggiore della soluzione ottimale vengono scartate (riga 1, algoritmo 3), invece se la soluzione corrente ha un livello inferiore, essa viene memorizzata come soluzione ottimale. Se i livelli della soluzione ottimale e della

soluzione corrente si equivalgono, si memorizza la soluzione corrente solo se essa ha un periodo maggiore della soluzione ottimale (riga 10, algoritmo 3).

Solo l'ultimo livello ha un criterio di selezione della soluzione ottimale differente, poiché si seleziona la soluzione con il minimo errore (riga 6, algoritmo 3). L'ultimo livello, rappresenta l'insieme dei valori d'errore più alti, che vanno da un valore fissato a $+\infty$, perciò all'interno di questo insieme illimitato, cerco di selezionare la soluzione che più si correla all'assegnamento dei punti.

Illustro i risultati dell'implementazione ed esecuzione del criterio degli standard, generando le soluzioni utilizzando il modello basato sul metodo dei vincoli, variando il minimo periodo β nel range fissato (sezione 3.2.2).

I livelli fissati sono

- Livello 1: errore $< 1e-5$
- Livello 2: errore $< 1e-2$
- Livello 3: errore $< 1e-1$
- Livello 4: errore < 1
- Livello 5: errore < 3
- Livello 6: errore < 5
- Livello 7: errore < 7
- Livello 8: errore < 10
- Livello 9: errore ≥ 10

Ho fissato i range in maniera da ottenere maggiore precisione e diversificazione delle soluzioni con errore tra 0 e 1, suddividendo tale spazio in 4 livelli. Nella definizione dei livelli, non ho tenuto in considerazione l'errore intrinseco dello strumento di rilevazione, per ottenere delle prime informazioni sulla prestazione dell'algoritmo in una situazione ideale. A seguito si elencano problemi con punti generati da una sinusoide $e(t)$ e nelle tabelle le soluzioni individuate dall'algoritmo.

I parametri dell'algoritmo 2 per le seguenti sperimentazioni sono:

- $f(x)$ in riga 9, algoritmo 1 pari alla funzione (8) con $k = 2$
- $periodoIniziale = 0s$
- $periodoFinale = 5000s$
- $tolleranzaDefault = 1e-6$

- $e(t) = \sin(0.0010t)$ sinusoide con periodo pari a: $T = \frac{2\pi}{\omega} = \frac{2\pi}{0.0010} = 6280s$

Tabella 11: periodo da individuare uguale a 6280s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	4830.7	3.2
20	4485.7	2.9
30	4186.6	0.6

- $e(t) = \sin(0.0013t)$ sinusoide con periodo pari a: $T = 4830.7s$

Tabella 12: periodo da individuare uguale a 4830.7s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	1495.2	2.9
20	4186.6	3.8
30	4186.6	0.6

- $e(t) = \sin(0.005t)$ sinusoide con periodo pari a: $T = 1256s$

Tabella 13: periodo da individuare uguale a 1256s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	1225.8	6.7
20	1395.5	4.7
30	4485.7	2.0

- $e(t) = \sin(0.0125t)$ sinusoide con periodo pari a: $T = 502.4s$

Tabella 14: periodo da individuare uguale a 502.4s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	697.8	3.6
20	785	11.6
30	514.7	0.6

- $e(t) = \sin(0.8t)$ sinusoide con periodo pari a: $T = 7.85s$

Tabella 15: periodo da individuare uguale a 7.85s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	7.85	0.6
20	7.85	80.0
30	7.85	4.2

L'esecuzione dell'algoritmo è caratterizzato da uno scostamento medio dal periodo da individuare di 937.2s e una mediana di 282.6s ed infine un tempo medio di esecuzione di 8.4s e una mediana di 3.2s.

È possibile provare a diminuire il tempo di esecuzione modificando l'algoritmo di controllo ed esplorazione, aggiungendo come condizione di aumento del passo, l'individuazione di una soluzione dal livello maggiore di un livello di soglia. Questo permette una velocizzazione dell'esplorazione saltando tutti quei periodi di partenza lontani dal periodo della sinusoide da individuare. Effettuando questa modifica, scegliendo ad esempio come livello di soglia il sesto, si ottengono i seguenti risultati:

- $e(t) = \sin(0.0010t)$ sinusoide con periodo pari a: $T = \frac{2\pi}{\omega} = \frac{2\pi}{0.0010} = 6280s$

Tabella 16: periodo da individuare uguale a 6280s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	4830.7	3.2
20	4830.7	3.9
30	4186.6	0.6

- $e(t) = \sin(0.0013t)$ sinusoide con periodo pari a: $T = 4830.7s$

Tabella 17: periodo da individuare uguale a 4830.7s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	1495.2	2.9
20	5709.0	7.3
30	4186.6	0.6

- $e(t) = \sin(0.005t)$ sinusoide con periodo pari a: $T = 1256s$

Tabella 18: periodo da individuare uguale a 1256s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	1495.2	5.9
20	1231.3	1.6
30	3140.0	10.4

- $e(t) = \sin(0.0125t)$ sinusoide con periodo pari a: $T = 502.4s$

Tabella 19: periodo da individuare uguale a 502.4s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	592.4	5.09
20	785	11.8
30	514.7	0.6

- $e(t) = \sin(0.8t)$ sinusoide con periodo pari a: $T = 7.85s$

Tabella 20: periodo da individuare uguale a 7.85s

Numero di punti	Periodo della soluzione (s)	Tempo di calcolo (s)
10	7.85	0.6
20	7.85	0.7
30	7.85	0.6

Così ottenendo uno scostamento medio di 825.5s e con mediana pari a 282.6s e un tempo medio di esecuzione di 3.7s con mediana pari a 2.9s. Si ottiene così una diminuzione sia dello scostamento medio che dei tempi di esecuzione.

Bibliografia

- [1] Steven G. Johnson, The NLOpt nonlinear-optimization package
- [2] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*, eds. S. Gomez and J.-P. Hennart (Kluwer Academic: Dordrecht, 1994), da pagina 51 a 67.
- [3] Dr. David R. Williams, Planetary Fact Sheets, NASA Goddard Space Flight Center 2016
- [4] Sartoretti P. and Schneide, J. "On the detection of satellites of extrasolar planets with the method of transits" 1999, da pagina 553 a 560, SAO/NASA Astrophysics Data System.