**----------------------------------- OPTIMIZATION AND GAME THEORY -----------------------------------**

**Set Convesso**

Il set è **convesso** se 🡺

**Set Affine**

Il set è **affine** se 🡺

**Combinazione Convessa**

**Lemma 1**

**SE** è convesso 🡺 ,

**Proposizione**

**SE**  rappresenta ogni possibile famiglia di set convessi 🡺 è convesso

Informalmente: l’**intersezione** di set convessi è anch’esso convesso

**Convex Hull**

di un set è il più piccolo set convesso che contiene

**Proposizione**

**Nota**

per convesso

**Definizione di Poliedro**

Un poliedro è l’intersezione di un numero finito di semipiani in

**Somma Di Set Convessi**

Siano e set convessi 🡺 è convessa (Vale anche per la differenza)

**Prodotto Di Set Convessi**

Sia un set convesso 🡺 è convesso

**Closure**

convesso 🡺 è convesso

**Interior**

convesso 🡺 è convesso

**Affine Set**

Sia 🡺è il più piccolo set affine che contiene

**Relative Interior**

Sia convesso 🡺

**Teorema**

Sia set convesso non vuoto 🡺 è convesso non vuoto

**Teorema**

Siano due set convessi non vuoti 🡺 sono linearmente separabili **SSE**

**Funzioni Affine**

Sia affine, ovvero con .

**SE** è convesso 🡺 è convesso

**SE**  è convesso 🡺 è convesso

**Definizione Di Cono**

Un set è un cono **SE** 🡺

**Cono Di Recessione**

Dato un poliedro il cono di recessione di è

**Cono Tangente •••**

Dato il cono tangente a è:

**Funzione Convessa**

Sia set convesso. Una funzione è convessa in se

**Teorema**

è convessa in **SSE**  è convesso

**Funzione Concava**

Sia set convesso. Una funzione è concava in se è convessa in

**Teorema**

è convessa in set convesso 🡺 continua nel

**Strictly Convex**

Dato set convesso, è **strictly** convex in se

**Strongly Convex**

Dato set convesso, è **strongly** convex in se :

**Teorema**

è strongly convex **SSE** è convessa

**Da cui segue che:**

è strongly convex **SSE** esiste una funzione convessa e :

**Teorema - Convex**

è convessa in **SSE**

**Teorema - Strictly Convex**

è strictly convex in **SSE**

**Teorema - Strongly Convex**

è strongly convex in **SSE**

**Teorema – Come capire i tipi di convessità •••**

è convessa in **SSE**  la Hessiana è **semi-definita positiva** ()

Ovvero che gli autovalori di sono:

• Se 🡺 è strictly convex

• è strongly convex in **SSE**  è **semi-definita positiva** . Ovvero:

Ovvero che gli autovalori di sono:

**Teoremi •••**

• **SE** è convessa e 🡺 è convessa

• **SE** sono convesse 🡺 è convessa

• **SE** è convessa 🡺 è convessa

• **SE** sono convesse 🡺 è convessa

• **SE** è una famiglia di funzioni convesse 🡺 è convessa

**Teoremi**

Siano e

• Se è convessa e è convessa non-decrescente 🡺 è convessa

• Se è concava e è convessa non-crescente 🡺 è convessa

• Se è concava e è concava non-decrescente 🡺 è concava

• Se è convessa e è concava non-crescente 🡺 è concava

**K-sublevel Set**

Data una funzione e dato definiamo il :

**Definizione – Quasi convex function**

Dato un set , la funzione si definisce **quasi convessa** in **SE** il set

È convesso .

**PROBELI DI OTTIMIZZAZIONE - INTRO**

**Problema Di Ottimizzazione**

Con

**Valore Ottimo**

Il valore ottimo di è definito come

**Ottimo Globale**

**Ottimo Locale**

**Teorema di Weierstrass •••**

Per continua in regione di ammissione chiusa e limitata 🡺 Esiste (almeno) un ottimo globale

**Corollario 2**

Se la funzione obiettivo è continua e la regione di ammissione è chiusa e tale che il : è non-vuoto e limitato 🡺 Esiste (almeno) un ottimo globale

**Funzione Coerciva**

**Corollario 3**

Se la funzione è **continua** e **coerciva**, e è chiuso 🡺 Esiste (almeno) un ottimo globale

**Teorema 1 •••**

Sia **convessa** in **convesso** 🡺 Ogni ottimo locale di è anche globale

**Proposizione 1**

Sia **strictly** **convex** in **convesso**, e ammette un ottimo globale 🡺 è l’unica soluzione ottima di

**Teorema 2**

è **strongly convex** in e è **chiuso** 🡺 Esiste un ottimo globale

**Corollario 1**

è **strongly convex** in e è **chiuso** e **CONVESSO** 🡺 Esiste un ottimo globale UNICO

**Teorema 3**

Sia **APERTO** e sia **differenziabile** in . Se è un ottimo locale per P 🡺

**Teorema 4 - Second order necessary optimality condition**

Sia aperto e ottimo locale per 🡺

**Teorema 5 – Second order sufficient optimality condition**

Sia aperto, preso assumiamo le seguenti condizioni vere

🡺 è un ottimo locale per

**Teorema 6 – Optimality condition for convex problems**

Sia differenziabile e convessa nel set convesso e aperto 🡺 è un ottimo globale per **SSE**

**Teorema 7**

Sia differenziabile e **STRICTLY CONVEX** nel set aperto e convesso 🡺 è un ottimo globale **UNICO** per **SSE**

**METODI DI OTTIMIZZAZIONE PER PROBLEMI NON VINCOLATI**

**Metodo del gradiente**

1. Scegliere ,
2. Step 2

**Proposizioni**

Sia continua e differenziabile

2. **SE** . Ovvero è un punto stazionario per

**Teorema**

Se è coerciva 🡺 punto di partenza della sequenza , essa è limitata, e ogni punto di accumulo di è un punto stazionario per

**Corollario**

Se è coerciva e convessa 🡺 punto di partenza della sequenza , essa è limitata, e ogni punto di accumulo di è un minimo globale per

**Corollario**

Se è stongly convex 🡺 punto di partenza della sequenza , essa converge all’unico minimo globale per

**Metodo del gradiente – Caso Quadratico**

Se , con .

Dove:

Per cui

**Metodo del gradiente con la linea di ricerca inesatta di Armijo**

1. , , ,
2. Step 2

**Metodo del gradiente coniugato**

1. Settare ,
2. Step 3

**Proposizioni**

• Un modo alternativo di calcolare

• Un modo alternativo di calcolare

• Se non troviamo il minimo globale dopo iterazioni 🡺 sono ortogonali

• Se non troviamo il minimo globale dopo iterazioni 🡺 sono coniugate rispetto a e è il minimo di su

**Teorema**

Il metodo del Gradiente Coniugato (CG) trova il minimo globale in, al massimo, iterazioni

**Teorema**

Se ha 🡺 CG trova il minimo globale in, al massimo, iterazioni

**Metodo di Newton – Basic Version**

1. Settare ,
2. Step 2

**Teorema Di Convergenza**

Se è un minimo locale di e 🡺 e la sequenza e

**Metodo di Newton – con linea di ricerca (inesatta)**

1. Settare , , ,
2. Step 2

**Teorema Di Convergenza**

Se è strongly convex 🡺 la sequenza minimo globale di

Se e 🡺 la convergenza è quadratica

**PROBLEMI VINCOLATI**

**Set Di Disequazioni Attive**

Definiamo il set di disequazioni attive

**First Order Feasible Direction Cone**

**Definizione – Abadie Constraint Qualification (ACQ)**

ACQ è valida per un punto ammissibile **SE**:

**Teoremi •••**

1. **Vicoli Affini**

**SE**  e sono funzioni affini 🡺 ACQ è verificata

1. **Slatex Condition**

**SE**  è convessa , e è affine , e

🡺 ACQ è verificata

1. **Dipendenza lineare dei gradienti dei vincoli attivi**

**SE**  e i vettori



sono linearmente dipendenti🡺 ACQ è verificata in

**TEOREMA KKT •••**

**SE**  è un minimo locale e **SE** l’ACQ è verificata 🡺 soluzione del sistema:

**LAGRANGIANA**

**KKT Con la lagrangiana**

Con , sostituibile con o

**Teorema – KKT per problemi convessi**

SE il problema P è convesso e è la soluzione del KKT 🡺 è un ottimo globale

**Rilassamento Lagrangiano**

**Funzione duale Lagrangiana**

1. Concava
2. per qualche punto
3. non differenziabile per qualche punto

**Teorema**

e 🡺 minimo globale per P

**Problema duale Lagrangiano**

D è per definizione **convesso** sempre.

**Teorema – Weak Duality**

Per qualsiasi problema di ottimizzazione P vale che:

**Teorema – Strong Duality**

Supponiamo che e siano continue e differenziabili:

con convesso, ottimo globale e l’è soddisfatto 🡺

1. La Strong Duality è verificata: e ammette l’ottimo
2. è l’ottimo per **SSE** è un vettore dei moltiplicatori KKT associato a

**Teorema**

è punto di **sella** per , ovvero

**SSE** è un ottimo per , e è un ottimo per , e vale

**SUPPORT VECTOR MACHINE**

**Iperpiano Di Separazione**

Dove e sono i due labeled-set. Affinché valga

**Funzione di Decisione**

🡸 classe

**Definizione – Margine di Separazione**

Se è l’iperpiano di separazione dei due insiemi e , allora definiamo il margine di separazione di come:

**Teorema**

Trovare l’iperpiano che ha il massimo margine di separazione equivale a risolvere il seguente problema di programmazione quadratico convesso:

**Linear SVM**

**Funzione Lagrangiana**

1. **SE** 🡺
2. **SE**  🡺 non dipende da ,

è strongly convex,

**Funzione Duale**

**Problema Duale**

Con e

1. Il duale è convesso:
2. ottimo del associato all’ottimo del primale , è ottimo anche del duale
3. **SE** 🡺 è detto **support vector**
4. **SE** è ottimo del duale 🡺
5. è ottenuto da: 🡺

**Linear SVM con SOFT MARGIN**

**Primale**

1. **SE** 🡺 il punto è miss-classificato e è un limite superiore al numero di punti miss-classificati
   1. **SE**  e 🡺 miss-classificato
   2. **SE**  e 🡺 miss-classificato

**Duale**

1. **SE** è una soluzione ottima di questo duale 🡺 =
2. Possiamo ottenere , fissando , e usando la condizione complementare (slackness): da cui:
3. è l’errore nel classificare il punto
4. **SE**  🡺
5. **SE**  🡺
6. **SE**  🡺 → condizione necessaria per **mis-classificare**

**Non-Linear SVM**

**Primale**

**Duale**

Troviamo , fissando :

**Funzione Kernel**

Definiamo , la funzione kernel:

**Teorema**

Se è una funzione Kernel, e se 🡺 la matrice

Possiamo usare la funzione per ridefinire il duale.

**In Pratica**

1. Scegliere una funzione Kernel
2. Trovare la soluzione ottima del duale
3. Scegliere , e trovare
4. Calcolare la decision function

**REGRESSIONE**

**Sistema per trovare il polinomio di regressione**

Dove: , ,

**Residual Vector**

Dove è la differenza fra valore reale e valore predetto.

**Polinomio di regressione con**

Con

Soluzione:

**Polinomio di regressione con**

Forma vettoriale:

Con:

**Polinomio di regressione con**

Forma vettoriale:

Con:

**Regressione Lineare**

Forma vettoriale:

Con:

**Regressione Lineare con variabili di rilassamento (Slack Variables)**

Forma vettoriale:

Con:

**Lagrangiana Lineare (Slack Variables)**

**Duale Lineare (Slack Variables)**

In formato vettoriale:

Con:

1. È un problema **convesso** di programmazione quadratica
2. **SE**  oppure 🡺 è un **support vector**
3. **SE** ( ottimo del duale 🡺
4. è ottenuto tramite la *complementary condition:* 
   * 1. 🡺
     2. 🡺

**Regressione NON Lineare**

Definiamo: , spazio delle caratteristiche, e cerchiamo una regressione lineare in nello spazio .

**Primale**

**Duale**

Ricordando che

**Passi**

1. Scegliere una funzione kernel:
2. Risolvendo il duale abbiamo: ( 🡺
3. *complementary condition:*
4. 🡺
5. 🡺

**Funzione di regressione per  NON Lineare**

**CLUSTER  
Modello**

Dove è la distanza fra il punto e il centroide

**Problema di Clustering**

1. **SE**  il problema è convesso
2. **SE**  il problema è non convesso e non differenziabile

**Teorema**

Il problema è equivalente al seguente problema non convesso e non differenziabile:

**K-Means**

1. Inizializzazione
   1. Set
   2. Setta i centroidi
   3. Setta
2. Aggiornamento centroidi
3. Aggiornamento clusters
4. Stopping condition
   1. Se 🡺 STOP
   2. Altrimenti e vai al passo 2

**Teorema**

L’algoritmo di K-Means termina dopo un numero finito di iterazioni in una soluzione del sistema del problema tale che

**Teorema**

Il problema è equivalente al seguente problema non convesso e non differenziabile:

**Teorema •••**

Il problema è equivalente al seguente problema non convesso e non differenziabile:

**K-Median**

1. Inizializzazione
   1. Set
   2. Setta i centroidi
   3. Setta
2. Aggiornamento centroidi
3. Aggiornamento clusters
4. Stopping condition
   1. Se 🡺 STOP
   2. Altrimenti e vai al passo 2

**Teorema**

L’algoritmo di K-Median termina dopo un numero finito di iterazioni in una soluzione punto stazionario tale che:

**METODI PER PROBLEMI DI OTTIMIZZAZIONE VINCOLATI**

**(Exact) Penality Method**

Trasformiamo il problema (convesso, e sono con le ) in un problema senza vincoli aggiungendo una penalità a

Dove

**Proposizione**

1. SE e sono continue e differenziabili 🡺anche lo è, e il gradiente è:
2. SE e sono convesse 🡺 anche lo è
3. Ogni problema ha una relazione con 🡺
4. SE risolve il problema 🡺 è ottimo anche per
5. SE 🡺

**Passi del metodo**

1. Settare
2. Settare
3. Settare
4. Trovare l’ottimo del problema
5. SE
   1. STOP
6. ALTIMENTI
   1. Passo 1

**Teoremi**

* **SE**  è coerciva 🡺 , sequenza limitata, e ogni punto stazionario è un’ottima soluzione di
* **SE**  🡺 è un’ottima soluzione di
* **SE**  e i gradienti dei vincoli attivi sono linearmente indipendenti 🡺 è un’ottima soluzione di e la sequenza vettore dei moltiplicatori del KKT associati a

(In quello esatto, dato che il problema convesso, abbiamo una sola soluzione e quindi un unico .)

**Barrier Method**

Questo metodo cerca una sequenza di punti ammissibili che approssima la soluzione ottima di , problema vincolato, approssimandolo con:

**Proprietà di**

1. convesso
2. derivabile con:

**Passi del metodo**

1. Settare
2. Settare
3. Settare
4. Scegliere
5. Settare
6. Trovare l’ottimo del problema
   1. Partendo da
7. **SE**
   1. **STOP**
8. **ALTIMENTI**
   1. Passo 1

**Trovare**

Consideriamo il problema ausiliare:

1. Prendiamo , e cerchiamo 🡺
2. Cercare la soluzione ottima dell’ausiliare: usando un Barrier Method
3. **SE** 🡺
   1. **ALTRIMETNI**

**PROBLEMI MULTI OBIETTIVO**

**Definizione**

**Ordine di Pareto**

Dati

1. Reflexive:
2. Asimmetrico: se e 🡺
3. Transitiva: se e 🡺

**Minimi di Pareto per un Sottoinsieme**

1. **Ideal Minimun**
   1. è un se
   2. è un se
2. **Minimun**
   1. è un se
   2. è un se
3. **Weak Minimun**
   1. è un se
   2. è un se

**Preposizione**

**Se**  🡺

**Teorema – Esistenza del minimo**

**Se**  è compatto 🡺

**Minimi di Pareto per un problema multi-obiettivo •••**

1. **Ideal Minimun**
   1. è un se ovvero se
2. **Minimun**
   1. è un se ovvero che
      1. Per qualche
3. **Weak Minimun**
   1. è un se ovvero che:

**Teorema •••**

**Se**  è continua e se è compatto 🡺

**Teorema •••**

**Se**  è continua e se è chiuso, e se

è un insieme non vuoto e limitato 🡺

**Corollario**

**Se**  è continua e se è chiuso, con coerciva per qualche

🡺

**Teorema**

è un **SSE** il problema ausiliario ha soluzione

**Teorema**

è un **SSE** il problema ausiliario ha soluzione

**PROBLEMI MULTI OBIETTIVO NON VINCOLATI**

**Definizione problema**

con continua e differenziabile

**Teorema**

**SE** è un **🡺** il problema non ha soluzione.

**Condizione Necessaria di Ottimalità**

**SE** è un 🡺 è una soluzione di:

**Condizione Sufficiente di Ottimalità**

**SE**  è convesso (Ovvero ogni è convessa) e è l’ottimo per  **🡺**

1. **SE** vale anche 🡺

**PROBLEMI MULTI OBIETTIVO VINCOLATI**

**Definizione problema**

e

**Teorema KKT Multi-Obiettivo**

**SE**  è un e **SE** l’ACQ è verificata in 🡺 è soluzione del sistema:

**Teorema – Condizione Necessaria di Ottimalità**

**SE** è un **🡺** il problema non ha soluzione.

**Corollario**

**SE**  è un e **SE** l’ACQ è verificata in il problema non ha soluzione.

**Teorema – Condizione Sufficiente di Ottimalità**

Assunti e sono convesse , e è affine :

1. **SE**  risolve il 🡺
2. **SE**  risolve il e 🡺

**Proposizione**

**SE**  è un unico minimo globale per in , per qualche 🡺

**SCALARIZATION METHOD**

Associamo al problema vincolato, il sistema che associa pesi a ogni :

Con insieme delle soluzioni ottime di

**Teorema**

**Teorema**

**SE** che è convesso e che sono convesse in , 🡺

**Teorema**

Sia è lineare (ovvero che è lineare ) e sia un poliedro 🡺

**Proposizione**

**SE**  è un unico minimo globale per , per qualche , 🡺

**NOTA**

Solo con si hanno , gli altri sono di Pareto; tranne che se con la funzione che rimane è strongly convex

**NOTA**

**SE** l’ottimo di coincide sia con che con 🡺 perché minimizza entrambe le funzioni obiettivo

**GOAL METHOD**

Definiamo nello spazio obiettivo un punto ideale , così definito:

**Teorema**

Per avvicinarsi il più possibile a , risolviamo:

• SE 🡺 ogni soluzione ottima di è un

• SE 🡺 ogni soluzione ottima di è un

**Goal Method – Norma 2**

**GAME THEORY – Matrix Game**

**Definizione two-person non-cooperative game**

**Equilibrio di Nash**

In un two-person non-cooperative game, una coppia di strategie è detta Equilibrio di Nash **SE**

**Matrix Game**

In un two-person non-cooperative game, dove:

* e sono set finiti:
* (zero-sum game)

definiamo la Matrix Game

**Definizione – Strategia Strictly Dominated**

Dato un two-person non-cooperative game, una strategia è strettamente dominata da se:

Similmente è strettamente dominata fa se:

Le strategie dominate possono essere eliminate dal gioco

**Mixed Strategies**

**SE**  è un Matrix Game 🡺

• una Mixed Strategy per il giocatore 1 è un vettore grande contenete probabilità.

• una Mixed Strategy per il giocatore 2 è un vettore grande contenete probabilità.

**Pure Strategy**

Sono i vertici di e :

**Expected Cost**

**Nota**

**Mixed Strategies Nash Equilibrium**

SE è un Matrix Game 🡺 è un Mixed Strategies Nash Equilibrium **SE**

O equivalentemente:

è un **punto di sella** per su

**Punto Di Sella**

Sia e . è un punto di sella della funzione **SE**

**Teorema**

soddisfa la condizione di punto di sella **SSE**

1. è una soluzione ottima di:
2. è una soluzione ottima di:

Tutte e tre le condizioni sono equivalenti a:

**Teorema – Esistenza del punto di sella**

Sia e . Assumendo che

1. e sono non vuoti, convessi e compatti
2. è continua e quasi-convessa in
3. è continua e quasi-convessa in

🡺 ammette un punto di sella in

**Corollario**

1. Ogni Matrix Game ha almeno un Mixed Strategies Nash Equilibrium
2. è un Mixed Strategies Nash Equilibrium **SSE**

Con valore ottimo pari a

**Teorema**

Il problema è equivalente a risolvere il seguente problema di programmazione lineare:

**Teorema**

Il problema è equivalente a risolvere il seguente problema di programmazione lineare:

**Proposizione**

è il duale di

**Risoluzione Esercizi**

I Pure Strategies Nash Equilibria possono essere cercati fra **i minimi di ogni colonna** della matrice per e **i massimi di ogni riga** per .

**I punti in comune sono Pure Strategies Nash Equilibria.**

**Nota**

**SE** il è convesso 🡺 Qualsiasi combinazione convessa fra due Pure Strategies Nash Equilibrium è una Mixed Strategies Nash Equilibria

**Definizione Combinazione Convessa con due punti**

**GAME THEORY – Bimatrix Games**

**Definizione**

Un Bimatrix Game è un two-person non-cooperative game, dove:

* e sono set finiti:
* (non zero-sum game)

**Teorema**

Ogni Bimatrix Game ha almeno un Mixed Strategies Nash Equilibrium

**Teorema**

Definendo e :

🡺 è un Nash Equilibrium **SSE**  e

**Teorema – KKT per Bimatrix Games**

è un Nash Equilibrium **SSE** :

È Verificato

**Proposizione**

è una soluzione per **SSE** è una soluzione ottima del problema di programmazione quadratica seguente:

E

**Formato Matriciale di**

Con: ()

**GAME THEORY – Convex Games**

Con continue e differenziabili

**Teorema**

**SE** le regioni di ammissione



Sono chiuse, limitate e convesse e le funzioni di costo:

1. è quasi-convessa
2. è quasi-convessa

🡺 esiste almeno un Nash Equilibrium

**Teorema – Condizione KKT**

**SE**  è un Nash Equilibrium e sono verificati sia in che in 🡺

:

**SE**  risolve il di sopra, e il game è convesso 🡺 è un Nash Equilibrium

**MATLAB**

**Funzione**

Dove. Il sistema che risolve, in generale è fatto:

**Funzione**

Dove. Il sistema che risolve, in generale è fatto:

**Funzione**

Dove. Il sistema che risolve, in generale è fatto:

function [c, ceq] = const(x)

c = […];

ceq = […];

end

**Funzione** uguale ma senza vincoli

**RISOLUZIONE ESERCIZI**

**ESERCIZIO 1: Mono-Objective**

1. **Provare se il problema ammette un ottimo globale**
   1. Controllare se è convessa tramite gli auto-vettori dell’Hessiana
   2. Controllare se è continua nel set di ammissione
2. **Applicare il metodo indicato tramite lo script MatLab**
3. **Il punto ottenuto è un minimo globale del problema?**
   1. Se è strongly convex allora il minimo globale è unico
   2. Per farlo bisogna vedere se la soluzione trovata è anche soluzione del associato al problema. Usando il teorema che dice: SE il problema è convesso e è la soluzione del KKT 🡺 è un ottimo globale

**ESERCIZIO 2: SVM**

1. **Scrivere il modello richiesto, copiandolo da sopra.**
2. **Risolvere usando MatLab, con i parametri descritti dalla traccia**
3. **Trovare i punti miss-classified**
   1. Bisogna considerare la “complementary slackness condition”, in particolare se il punto è miss-classificato allora l’errore , per cui da deriviamo . Cercare i punti con questo . Ma non basta, potrebbero esserci punti classificati correttamente anche con . Per notarlo basta calcolare il valore dell’Iperpiano con il punto indicato, e vedere se soddisfa:
4. **Classifica il nuovo punto**
   1. Bisogna vedere se:

**ESERCIZIO 2: Regressione**

1. **Scrivere il modello richiesto, copiandolo da sopra.**
2. **Risolvere usando MatLab, con i parametri descritti dalla traccia**
3. **Trovare i Support Vector:**
   1. Usare MatLab, i SV sono dati dalla seconda e dalla terza colonna
4. **Trovare i punti che cadono fuori dal usando la soluzione del duale**
   1. Guardando le condizioni del primale:

Se un punto è fuori dal , allora l’errore è

Bisogna cercare i punti a cui corrisponde un errore positivo, cerchiamo prima nella soluzione duale i punti , e cerchiamo i punti che hanno , per lo stesso principio del problema .

**ESERCIZIO 2: Clustering**

1. **Scrivere il modello richiesto, copiandolo da sopra.**
2. **Risolvere usando MatLab, con i parametri descritti dalla traccia**

**ESERCIZIO 3: Multi-Objective**

1. **Prova l’esistenza del Minimo di Pareto, trovare i e i** 
   1. Per farlo risolvere il dello scalarizied problem, attraverso il , provando che il problema è convesso e che le sono soddisfatti:
      1. Per
      2. Per
      3. Per
         1. Distinguere i casi per e
   2. **SE** il problema è lineare allora tutti i risultati trovati sono o secondo il teorema di sopra descritto
   3. **SE** una delle è Strongly Convex allora ha un unico minimo che è un

**ESERCIZIO 4: Mono-Matrix**

1. **Trovare le strategie dominate per eliminarle:**
   1. Player 2: Eliminiamo le **colonne** che hanno tutti i valori agli elementi corrispondenti delle altre colonne (Mi tengo la colonna maggiore)
   2. Player 1: Eliminiamo le **righe** che hanno tutti i valori agli elementi corrispondenti delle altre righe (Mi tengo la riga minore)
2. **Trovare le Pure Strategies Nash Equilibiria**
   1. Cercare gli elementi che sono contemporaneamente **massimi per la propria riga** e **minimi per la propria colonna**.
   2. Ricordare di mantenere gli indici della matrice originale
3. **Trovare un Mixed Strategies Nash Equilibiria**
   1. Risolvere il sistema primale dato dalla definizione sopra, con MatLab
   2. **SE** il risultato del sistema è un Pure Strategies Nash Equilibiria, allora controllare se il problema è lineare, per cui ogni combinazione convessa fra due punti di Pure Strategies Nash Equilibiria è un Mixed Strategies Nash Equilibiria.

**ESERCIZIO 4: Bi-Matrix**

1. **Fa tutto MatLab** 😂