positioning

[program=makeindex,columns=2,intoc=true,options=-s ../../general/pyro.ist] firstpagestyle=empty, othercode=

|||||

todolistitemize2 [todolist]label=

–[[Require library for Lua library]] require("lualibs.lua")

function tableMerge(t1, t2) for k,v in pairs(t2) do if type(v) == "table" then if type(t1[k] or false) == "table" then tableMerge(t1[k] or , t2[k] or) else t1[k] = v end else t1[k] = v end end return t1 end

–[[Opens the two metadata file]] local specificFile = io.open('metadata.json') local folderFile = io.open('../metadata.json') local genericFile = io.open('../../metadata.json')

–[[Reads the files]] local specificJsonString = specificFile:read('*a') local folderJsonString = folderFile:read('*a') local generalJsonString = genericFile:read('*a')

–[[Closes the files]] specificFile.close() folderFile.close() genericFile.close()

–[[Convert the Json strings in Lua dictionaries]] local specificJson = utilities.json.tolua(specificJsonString) local folderJson = utilities.json.tolua(folderJsonString) local generalJson = utilities.json.tolua(generalJsonString)

–[[Merge top layer of dictionaries, so that the specific one overrides the generic one.]]

metadata = tableMerge(tableMerge(generalJson, folderJson), specificJson)

if true then tex.print("
input
main/../../general/italian.tex") else tex.print("
input
main/../../general/english.tex") end

folFOLFirst Order Logic

–[[Load data into variables to simplify code afterwards]] title = metadata["title"] cfu = metadata["cfu"] year = metadata["year"] degree = metadata["degree"] university = metadata["university"] notesType = metadata["notesType"] professors = metadata["professors"] authors = metadata["authors"]

tex.print("
MakeUppercase
textbf
large"..title.."")

for key, value in pairs(professors) do tex.print(’Prof. ’..value["name"] .. " " .. value["surname"].."
") end   if cfu > 0 then tex.print(cfu.." CFU
") end

for key, value in pairs(authors) do tex.print(’
textbf’..value["name"] .. " " .. value["surname"].."
") end

tex.print(notesType.."
")   tex.print(year.."
")

tex.print(degree.."
")   tex.print(university.."
") Italy
October 16, 2017

TITLEPAGE NOT RENDERED!
RECOMPILE WITH LUATEX!

Contents

Chapter 1

Introduction

1.1 Dispense

Sono disponibili dispense sul sito del corso.

Chapter 2

Problemi di Decisione

2.0.1 Problemi complessi

$$P = (X, \Omega, F, f, D, \Pi)$$

Figure 2.1: Definizione formale di problema di decisione.

Queste variabili rappresentano:

1. X rappresenta l'insieme delle **alternative**, o delle **soluzioni** o anche delle **soluzioni ammissibili**.
2. Ω rappresenta insieme degli **scenari** o **esiti**.
3. F rappresenta l'insieme degli **impatti**.
4. f rappresenta la **funzione dell'impatto**.
5. D rappresenta l'insieme dei **decisori**, tipicamente un insieme finito e di dimensione bassa. Un decisore è un'entità umana, modellata quanto possibile matematicamente.
6. Π insieme delle **preferenze**.

X viene definito come:

$$X \subseteq R^n \text{se } \boldsymbol{x} \in X \Rightarrow \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

con ogni termine x_i viene chiamato o **elemento di alternativa** o **variabile di decisione**.

Ω viene definito come:

$$\Omega \subseteq R^r \text{se } \boldsymbol{\omega} \in \Omega \Rightarrow \boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \dots \\ \omega_r \end{bmatrix}$$

con ogni termine ω_i viene chiamato o **elemento di scenario** o **variabile esogene**, cioè variabili che influiscono sulla configurazione del nostro sistema, non decise arbitrariamente ma provenienti dall'esterno.

F viene definito come:

$$F \subseteq R^p \text{se } \boldsymbol{f} \in F \Rightarrow \boldsymbol{f} = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_p \end{bmatrix}$$

Le $f_l \in R$ vengono ipotizzate ad essere intere e vengono chiamate **indicatore**, **attributo**, **criterio** o **obbiettivo**. Un **indicatore** per esempio potrebbe essere un *valore ottimo*.

La f viene definita come:

$$f(\boldsymbol{x}, \boldsymbol{\omega}) : X \times \Omega \to F$$

La matrice di tutte le combinazioni viene chiamata **matrice delle valutazioni**.

La Π viene definita come

$$\Pi : D \to 2^{F \times F}$$

, dove $\pi_d \subseteq F \times F$. $F \times F$ rappresenta l'insieme delle **coppie ordinate di impatti**, mentre $2^{F \times F}$ rappresenta l'insieme delle **relazioni binarie**.

Per esempio, ponendo $F = \{f, f', f''\}$, otteniamo un prodotto cartesiano:

$$F \times F = \{(f, f'), (f, f''), (f', f), (f, f''), (f'', f), (f'', f'), (f, f), (f', f'), (f'', f'')\}$$

La **preferenza** è la volontà per cui il decisore risulta disponibile a fare uno scambio.

Un esempio di preferenza è:

$$f'_d f' \Leftrightarrow (f', f'') \in \Pi_d$$

. In un ambiente ingegneristico si usa il $_d$, minimizzando i costi, mentre in un ambiente economico si cerca di massimizzare i costi $_d$.

Definizione 2.0.1 (indifferenza) *Due preferenze f' e f'' sono dette* ***indifferenti*** *quando:*

$$f' \ f'' \Leftrightarrow \begin{cases} f'_d f'' \\ f'_d f'' \end{cases}$$

Definizione 2.0.2 (Preferenza Stretta) *Una preferenza f' è detta* ***preferenza stretta*** *quando:*

$$f' <_d f'' \Leftrightarrow \begin{cases} f'_d f'' \\ f'_d f'' \end{cases}$$

Definizione 2.0.3 (Incomparabilità) *Due preferenze f' e f'' sono dette* ***incomparabili*** *quando:*

$$f'_d f'' \Leftrightarrow \begin{cases} f'_d f'' \\ f'_d f'' \end{cases}$$

2.0.2 Proprietà delle preferenze

Proprietà riflessiva

$$f f \qquad \forall f \in F$$

Proprietà di completezza

Un decisore può sempre concludere una decisione (ipotesi molto forte che talvolta porta a risultati impossibili):

$$f f' \Rightarrow f' f \qquad \forall f, f' \in F$$

Proprietà di anti-simmetria

$$f f' \wedge f' f \Rightarrow f' = f \qquad \forall f, f' \in F$$

Proprietà Transitiva

Solitamente i decisori non possiedono questa proprietà, anche perché è necessario modellare lo scorrere del tempo, per cui le proprietà valgono potenzialmente solo in un determinato periodo temporale. Viene generalmente considerata verificata.

$$f f' \wedge f' f'' \Rightarrow f f'' \qquad \forall f, f', f'' \in F$$

2.0.3 Ipotesi funzione del valore

Un decisore che ha in mente una funzione valore v, ha in mente una relazione di preferenza Π **riflessiva**, **completa**, **non necessariamente anti simmetrica** e **transitiva**. Quando una relazione possiede queste proprietà viene chiamata **ordine debole**, debole perché possono esiste dei *pari merito*. Un campo di applicazione sono i campionati sportivi.

$$\exists v : F \to R : ff' \Leftrightarrow v_{(f)} v_{(f')}$$

Condizioni di preordine

Avendo le condizioni di **riflessività**, **transitività** si ottiene la condizione di **preordine**.

Ordini deboli

Avendo le condizioni di **riflessività**, **transitività** e **completezza** si ha la condizione di ordine debole, che è molto utilizzata.

Ordine parziale

Avendo le condizioni di **riflessività**, **transitività** e **antisimmetria** si ottiene la condizione di **ordine parziale**.

Ordine totale

Avendo le condizioni di **riflessività**, **transitività**, **completezza** e **antisimmetria** si ottiene la condizione di **ordine totale**.

2.0.4 Tabella riassuntiva

Proprietà	**Preordine**	**Ordine debole**
Riflessività	[scale=0.4](0,.35) − (.25,0) − (1,.7) − (.25,.15) − cycle;	[scale=0.4](0,.35) − (.25,0) − (1,.7) − (.25,.15) −
Transitività	[scale=0.4](0,.35) − (.25,0) − (1,.7) − (.25,.15) − cycle;	[scale=0.4](0,.35) − (.25,0) − (1,.7) − (.25,.15) −
Completezza		[scale=0.4](0,.35) − (.25,0) − (1,.7) − (.25,.15) −
Antisimmetria		

2.1 Conto di Borda

La formula in figura borda utilizzato per costruire una **funzione valore**:

$$v(f) = \{f' \in F : ff'\}$$

Figure 2.2: Conto di Borda

Il valore di un impatto è pari al numero di impatti cui esso è preferibile, compreso l'impatto stesso.

Quando la cardinalità dell'insieme è $N \times R$ è possibile ottenere una **funzione valore**, ma quando ci si trova in condizioni come $R \times R$ che non risultano più mappabili sull'insieme R non risulta più possibile realizzare una **funzione valore**.

2.2 Problemi semplici

Un problema viene detto *semplice* quando essi possiedono queste caratteristiche:

1. $\exists v(f)$ conforme
2. $\Omega = 1 \Rightarrow f : X \to R$, cioè esiste un $f(x)$
3. $D = 1$
4. $X = \{x \in R^n : g_j(x) \leq 0 \forall j = 1, ..., n\}$ con $g_j \in C^1(R^n)$

Chapter 3

Programmazione matematica

Minimizzo $f(x)$, con la condizione di $g_j(x) \leq 0 \forall j = 1...n$.

Supponiamo di voler identificare la posizione migliore di una discarica, e che il punto in cui i rifiuti vengono prodotti sia $R = (1, 0)$, che in punto $C = (0, 0)$ vi sia in una città e che si debba avere una distanza di almeno 2 dalla città. Inoltre, la nostra discarica deve trovarsi a sinistra di $\frac{3}{2}$, cioè $x_0 < \frac{3}{2}$, perché li vi è un confine.

[->] (-1,0) – (3,0) node[right] x_1; [->] (0,-1) – (0,3) node[above] x_2;

/in (1,0)/R, (0,0)/C [fill=black] circle (0.05) node[above right] ;

[domain=-1:3,smooth,variable=,red] plot (3/2,);

La funzione di minimo che vado a definire risulta:

$$\min f(x) = \begin{cases} (x_1 - 1)^2 + x_2^2 \\ x_1^2 + x_2^2 = 4 \\ x_1 < \frac{3}{2} \end{cases}$$

[grid=major, samples=80, xlabel=x_1, ylabel=x_2, zlabel=$fitness$]

3[surf, unbounded coords=jump] x$^2 + y^2 > 4x < 3/2?(x-1)^2 + y^2 : NaN$;

3.1 Programmazione matematica

Definizione 3.1.1 *Ottimo locale* $\widetilde{x}$ *ottimo locale* $\Leftrightarrow f(x) \geq f(\widetilde{(x)}) \forall x \in U_{\widetilde{x},\epsilon}$

Dato $\widetilde{x}$ come un **ottimo locale**, e $\xi(\alpha)$ un **arco ammissibile** con la caratteristica di:

$$\xi(0) = \widetilde{x} \qquad \xi(alpha) \in X \forall \alpha \in [0, \widehat{\alpha})$$

Allora vale che ξ risulta **non migliorante**:

$$f(\xi(\alpha)) \geq f(\widetilde{x}) = f(\xi(0)) \forall \alpha \in [0, \widehat{\alpha})$$

La formula sovra riportata può essere espressa più semplicemente tramite:

$$[\nabla f(\widetilde{x})]^T P_\xi \geq \emptyset$$

Definizione 3.1.2 (Punti non regolari)

$$\widetilde{x} \textit{ regolare } \Leftrightarrow \nabla g_j(\widetilde{x}) \textit{per } g_j \textit{ attivo, con le varie funzioni } g_j \textit{ linearmente indipendenti}$$

Definizione 3.1.3 (Punti non regolari) *Sono dei punti per cui non vale*

$$[\nabla g_j(\widetilde{})]^T P_\xi(\widetilde{x}) \geq 0 \ \textit{per} \ g_j \ \textit{attivo} \ \Leftarrow \begin{cases} \xi \ \textit{arco ammissibile} \\ \widetilde{x} \ \textit{ottimo locale} \end{cases} \Rightarrow [\nabla f(\widetilde{x})]^T P_\xi \geq \emptyset$$

[grid=major, samples=80, xlabel=x_1, ylabel=x_2, zlabel=$fitness$]

3[surf, unbounded coords=jump] (x-1)$^3 + y < 0((x-1)^3 - y < 0)?1 : 0;$

3.2 Lemma di Farkas

Non ho capito a che serve

$$C_j = \{p \in R^2 : g_j^T p \leq 0 \forall j\}$$

Figure 3.1: Cono direzioni "opposte" ai vettori g_j

$$C_f = \{p \in R^2 : f^T p \leq 0 \forall j\}$$

Figure 3.2: Cono direzioni "opposte" a f

Se $\exists \mu_j \geq 0 : f = \sum_j \mu_j g_j \Leftrightarrow (C_g \subseteq C_f) - f^T p \leq 0 \forall p : g_j^T p \leq 0 \forall j$
Posso riscrivere questa formula usando i gradienti:
Se $\exists \mu_j \geq 0 : \nabla f = \sum_j \mu_j \nabla g_j \Leftrightarrow (C_g \subseteq C_f) - \nabla f^T p \leq 0 \forall p : \nabla g_j^T p \leq 0 \forall j$
che cosa è la combinazione lineare? e convessa? e conica?

3.3 Altra roba che non capisco

Se $\widetilde{(x)}$ è un **ottimo locale** e **regolare**, allora $\exists \mu_j \geq 0 : \nabla f(\widetilde{x}) + \sum_{j:g_j \text{ attivo}} \mu_j \nabla g_j = \emptyset$
Questo viene posto a sistema con $\mu_j g_j(\widetilde{x}) = \emptyset \forall j = 1...n$:

$$\begin{cases} \exists \mu_j \geq 0 : \nabla f(\widetilde{x}) + \sum_{j:g_j \text{ attivo}} \mu_j \nabla g_j = \emptyset \\ \mu_j g_j(\widetilde{x}) = \emptyset \forall j = 1...n \\ g_j(\widetilde{x}) < 0 \Rightarrow \mu_j = 0 \\ g_j(x) \leq 0 \forall j = 1...m \end{cases}$$