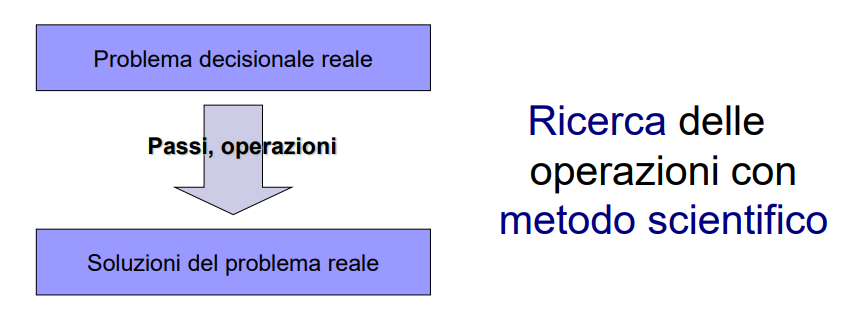
04/10/2022: Introduzione al corso

Ricerca Operativa (Operations Research) 🡪 Supporto ai processi decisionali in sistemi complessi, operando la decisione migliore a seconda del tipo di problema decisionale.

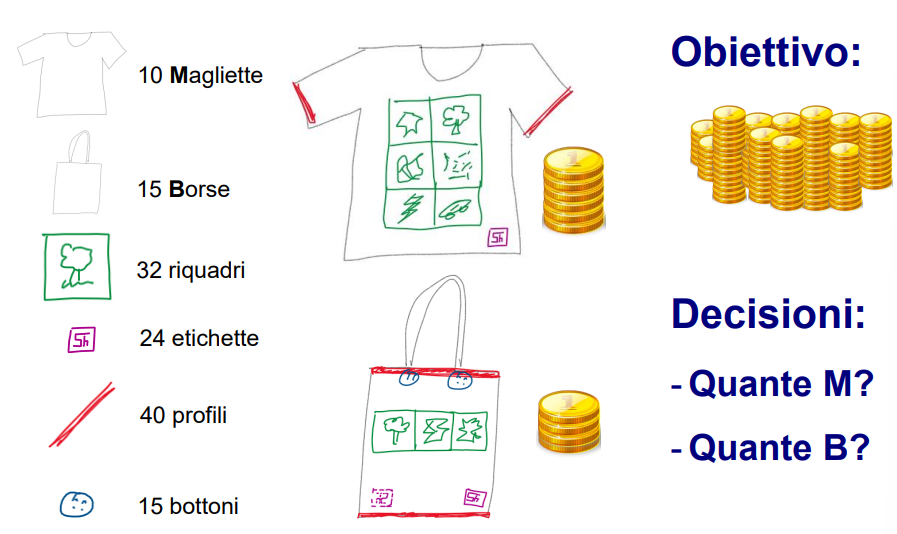
Cerco i passi che mi permettono di passare dalla decisione del problema alla soluzione stessa, applicando dei passi (quindi, un algoritmo), cercando un’ottimizzazione (soluzione migliore possibile)



Da Wikipedia, definizione espansiva e quella presente alla data di scrittura di questo file:

*“La ricerca operativa (nota anche come teoria delle decisioni, scienza della gestione o, in inglese, operations research ("Operational Research" in Europa) e indicata con le sigle RO o OR) è la branca della matematica applicata in cui problemi decisionali complessi vengono analizzati e risolti mediante modelli matematici e metodi quantitativi avanzati (ottimizzazione, simulazione, ecc.) come supporto alle decisioni stesse. La ricerca operativa riveste un ruolo importante nelle attività decisionali perché permette di operare le scelte migliori per raggiungere un determinato obiettivo rispettando vincoli che sono imposti dall'esterno e non sono sotto il controllo di chi deve compiere le decisioni.*

*L'obiettivo è dunque quello di fornire un supporto alla presa di decisioni. Per giungere a questo scopo, la ricerca operativa fornisce strumenti matematici e algoritmici di supporto alle attività decisionali in cui occorre gestire e coordinare attività e risorse limitate al fine di massimizzare o minimizzare una funzione obiettivo.*

*La ricerca operativa si occupa dunque di formalizzare un problema in un modello matematico e calcolare una soluzione ottima, quando possibile, o approssimata (detta anche subottima) per esso. Essa costituisce un approccio scientifico alla risoluzione di problemi complessi, si può ricondurre all'ambito della matematica applicata, ma presenta forti caratteristiche interdisciplinari relative in prevalenza a matematica, informatica, economia e finanza, ingegneria ed altre. Inoltre, la ricerca operativa ha molte applicazioni commerciali soprattutto negli ambiti economico, infrastrutturale, logistico, militare, della progettazione di servizi e di sistemi di trasporto e nelle tecnologie. Nel caso particolare di problemi di carattere economico, la funzione da massimizzare può coincidere con il massimo profitto ottenibile o con il minor costo da sostenere.”*

Esempio pratico di ottimizzazione: creazione di magliette e borse con un certo numero di riquadri.

L’obiettivo sarebbe di creare un programma per massimizzare il profitto date queste risorse.

Serve un *criterio* per capire come ragionare sulle risorse. Per esempio, posso ragionare ordinando in base al profitto, sommando le risorse che si devono usare e dividendo per quantità.

Altri ragionamenti possibili:

* scelte greedy (migliori sulla base di un input) sulla base delle magliette e delle borse
* bruteforce/approccio esaustivo-esponenziale (provo tutte le combinazioni possibili)
* scelgo una quantità casuale e cerco di vedere quale tra tutte si avvicina di più al valore che mi serve

Problemi di ottimizzazione

Determinare la migliore configurazione di sistemi complessi sotto condizioni di utilizzo di risorse scarse

* Mix ottimo di produzione
* Pianificazione della produzione, schedulazione di processi
* Determinazione dei turni del personale
* Determinazione di percorsi ottimali
* Organizzazione dei flussi di dati in una rete di telecomunicazione
* Individuazione di sequenze genomiche
* Pianificazione e gestione operativa di reti di trasporto (Amazon)

Si cerca in effetti di:

* Generare delle soluzioni ammissibili, quindi valide nel contesto d’uso
* Proporre delle soluzioni ragionevoli, quindi contestualmente utili nel contesto d’uso

Esempi utili e reali in merito (approfondimenti messi a scopo didattico)

* *Problema del commesso viaggiatore – Traveling salesman problem*

Il problema è NP-Completo, quindi un problema che è possibile verificare in tempo polinomiale (cioè, verificare la validità dell’ipotesi), ma non risolvibile nello stesso tempo, risultando un problema complesso.

Ci troviamo di fronte ad un quesito assai naturale: data una serie di città, qual è il modo più veloce (ovvero la strada più corta) per un commesso viaggiatore di visitare tutte le città una e una sola volta, e ritornare da dov’è partito?

È un problema complesso da risolvere; al crescere del numero di possibili città, il problema potrebbe non avere mai soluzione. Una soluzione venne dal prof. Christofides, il quale creò l’albero più corto che connette tutte le città, rete che connette tutte le città senza creare cicli. Se il percorso esiste, ogni città deve avere un numero pari di strade che la raggiungono. n altre parole, per ogni città, ogni arrivo è seguito da una partenza, il che significa che ci serve un numero pari di archi che tocchino la città nel network che costruiamo. La buona notizia è che anche l’inverso è vero: in un network in cui ogni città ha un numero pari di strade che la toccano, esiste un percorso che passa per ogni città una e una sola volta e torna alla città di partenza.

L’algoritmo, partendo dal network senza cicli, va ad aggiungere archi alle città con un numero dispari di strade, fintanto che ne esistono. Il percorso che ne esce alla fine non sarà necessariamente il migliore che il nostro commesso viaggiatore possa scegliere, ma non si allontana troppo dalla migliore scelta. Infatti, Christofides dimostrò che il suo metodo crea nel peggiore dei casi un percorso che è al massimo il 50% più lungo del percorso ottimo.

Esiste un nuovo algoritmo trovato dopo circa 50 anni, comunque molto simile a quello di Christofides. La differenza fondamentale consiste nel fatto che l’albero da cui si parte non è più il più corto possibile, ma è creato per via di un processo probabilistico e facendo in modo che le città con un numero dispari di strade siano vicine tra loro. Dopodiché l’algoritmo riprende l’originale e aggiunge strade a queste città fino ad ottenere un loop completo.

* Problema dello zaino – Knapsack problem

Nel problema dello zaino, è necessario imballare un insieme di oggetti, con valori e dimensioni date (come pesi o volumi), in un contenitore con una capacità massima. Se la dimensione totale degli oggetti supera la capacità, non è possibile imballarli tutti. In questo caso, il problema è scegliere un sottoinsieme di oggetti di valore totale massimo che possa entrare nel contenitore.

Esistono due tipi di problemi knapsack:

* 0/1 Knapsack Problem

Significa che gli oggetti o sono completamente o non sono riempiti in uno zaino (quindi, o li metti interi o non li metti). Ad esempio, abbiamo due oggetti che pesano rispettivamente 2 kg e 3 kg. Se prendiamo l'oggetto da 2 kg, non possiamo prendere un oggetto da 2 kg (l'oggetto non è divisibile); dobbiamo prendere completamente l'oggetto da 2 kg.

Il problema si risolve con la programmazione dinamica. Per risolverlo con la programmazione dinamica, prendiamo pesi e valori degli oggetti, cercando di assegnare, risolvendo i sottoproblemi in una matrice unica (quello che fa la programmazione dinamica), tenendo conto delle assegnazioni di quantità e pesi rispettive nelle varie applicazioni.

In questo modo, sappiamo caso per caso se riusciamo ad assegnare pesi e quantità.

* Fractionary knapsack problem

Significa che possiamo dividere l'oggetto. Ad esempio, se abbiamo un oggetto di 3 kg, possiamo prendere l'oggetto di 2 kg e lasciare quello di 1 kg. Il problema dello zaino frazionato si risolve con l'approccio Greedy. L'idea di base dell'approccio greedy consiste nel calcolare il rapporto valore/peso per ogni elemento e nell'ordinare gli elementi in base a questo rapporto. Quindi si prende l'elemento con il rapporto più alto e lo si aggiunge fino a quando non è possibile aggiungere l'elemento successivo nella sua interezza e alla fine si aggiunge l'elemento successivo il più possibile.

Però:

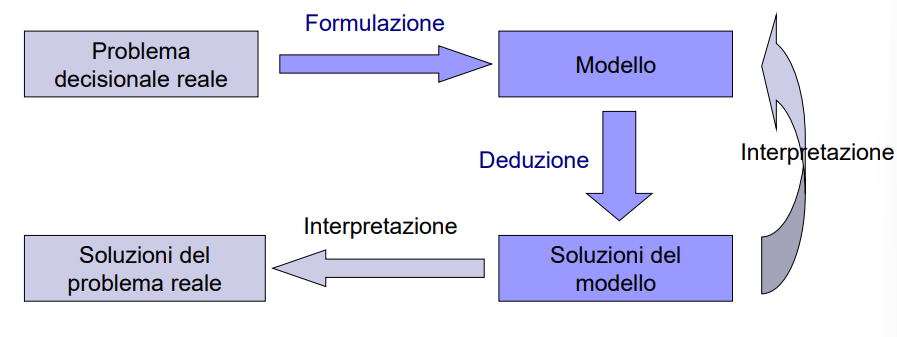
◼ Come certificare che una soluzione proposta è la migliore in assoluto (ottima)?

◼ Come valutare il valore intrinseco (significato qualitativo) delle risorse (e.g. un riquadro)

◼ Come valutare la stabilità della soluzione proposta in funzione di variazioni dei dati (ricavi, risorse disponibili etc.)?

◼ Come stabilire le soluzioni ottime in problemi simili (prospettiva modellistica e algoritmica)?

Per risolvere questo, si ha l’uso di strumenti matematici e algoritmici (teorema e dimostrazione), tramite la Ricerca Operativa, certificando la soluzione ottima.

Il metodo della Ricerca Operativa è basato su:

◼ Formulazione: modelli matematici, su grafo, di simulazione, teoria dei giochi, data-driven

(intelligenza artificiale)

◼ Deduzione: metodi quantitativi, algoritmi efficienti.

Nell’uso di un modello, non si trova una soluzione del problema; questo ovviamente ha dei limiti, in quanto la soluzione deve essere interpretata sulla base della realtà (se è un buon modello, si adatta bene).

Vediamo un esempio reale per capire l’utilizzo della matematica:

*“Un coltivatore ha a disposizione 11 ettari di terreno da coltivare a lattuga o a patate. Le risorse a sua disposizione, oltre al terreno, sono: 70 kg di semi di lattuga, 18 t di tuberi, 145 mc di fertilizzante. Supponendo che il mercato sia in grado di assorbire tutta la produzione e che i prezzi siano stabili, la resa stimata per la coltivazione di lattuga è di 3000 €/ettaro e quella delle patate è di 5000 €/ettaro. Il consumo di risorse per ogni tipo di coltivazione è di 7 kg di semi e 10 mc di fertilizzante per ettaro di lattuga, e 3 t di tuberi e 20 mc di fertilizzante per le patate. Stabilire quanto terreno destinare a lattuga e quanto a patate in modo da massimizzare la resa economica e sfruttando al meglio le risorse disponibili.”*

Dati utili:

* Lattuga e Patate (quali risorse)
* Semi, Tuberi e Fertilizzante (quali sottotipi di risorse e incrocio possibile dei dati)

Di fatto, in pratica, è come il problema di prima.

Non si può adottare un approccio forza bruta (quantità continue e non discrete). In realtà, questo ci aiuta.

*Costruzione del modello*

◼ Cosa bisogna decidere? ⇒ variabili decisionali (incognite che rappresentano i dati)

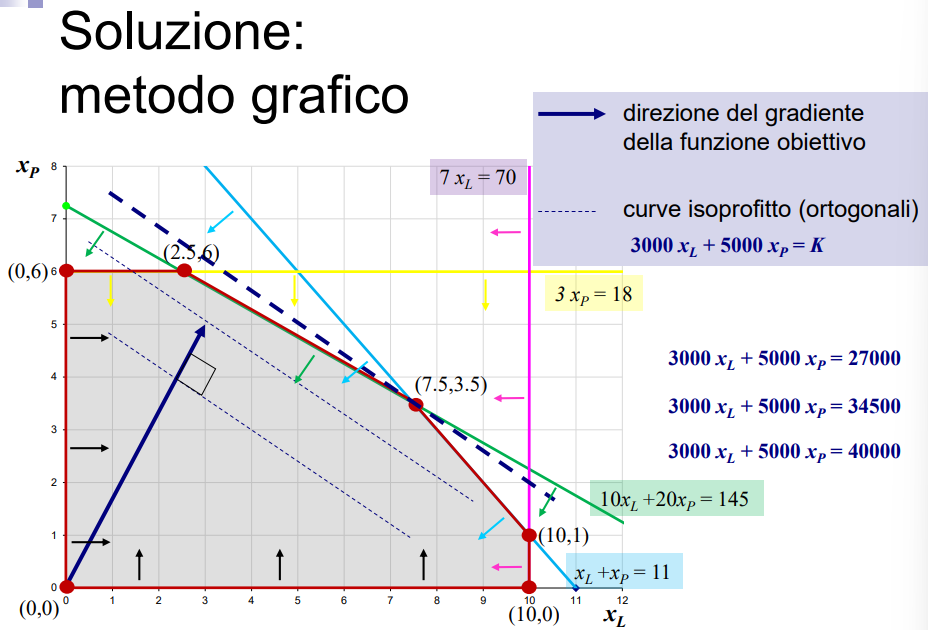
◼ Qual è l’obiettivo? ⇒ funzione obiettivo (ciò che vogliamo ottenere, mette insieme le v. decisionali)

◼ Come sono caratterizzate le soluzioni ammissibili? ⇒ vincoli del problema (relazioni tra incognite)

Immagine che contiene testo

Descrizione generata automaticamente◼ Modelli matematici ⇒ f. obiettivo e vincoli visti come relazioni matematiche tra le variabili decisionali

Aggiungiamo che quelle citate sotto come curve isoprofitto sono la differenza tra il ricavo ed i costi totali.



Per trovare la soluzione ottima, usiamo la funzione obiettivo.

Man mano che aumenta il guadagno, ho sempre più rette che vanno verso il *gradiente* (piano delle soluzioni ammissibili in rosso, possiamo dire il vettore che descrive il “movimento” delle derivate parziali delle rette); finché ci sono rette parallele, aumento il profitto, fino ad arrivare al limite della regione.

La soluzione ottima qui è 40000, in cui la soluzione ottima è il punto (7.5, 3.5).

Il metodo grafico funziona grazie a due motivi:

⇒ la linearità della funzione obiettivo

⇒ la linearità dei vincoli

Si parla in questi casi di modelli di programmazione lineare (PL) (come in questo caso, almeno due variabili e funzioni isoprofitto sotto forma di rette).

Sotto queste ipotesi (come vedremo meglio in seguito), una soluzione si trova su un vertice della regione ammissibile: l’ultimo toccato traslando le rette isoprofitto nella direzione del gradiente.

Se ci sono più variabili, si cerca di passare ad un problema a livello geometrico, tale che si possa approssimare comunque algebricamente il problema.

*Programma di massima del corso:*

1. Problemi di ottimizzazione: modellazione e soluzione con software off-the-shelf (app già pronte da usare)

- formulazione di modelli di programmazione matematica;

- soluzione con l’utilizzo di pacchetti software (laboratorio).

2. Programmazione lineare:

- teoria e metodo del simplesso;

- teoria della dualità e applicazioni.

3. Ottimizzazione su grafi: modelli e algoritmi per

- problema del cammino minimo;

- problemi di flusso su reti (flusso massimo, flusso di costo minimo).

4. Introduzione alla Programm. Lineare Intera e all'Ottimizzazione Combinatoria

- metodo del Branch & Bound per PL;

- cenni sui metodi euristici e metaeuristici (ricerca locale e varianti)