1. Come riconosco dal tableau che mi trovo in una soluzione di base degenere?

Il tableau è la rappresentazione sotto forma di tabella di un problema di programmazione lineare. Al suo interno possiamo identificare le soluzioni di base, ovvero le soluzioni del sistema dopo aver fissato a zero le n variabili fuori base. Una soluzione di base viene definita degenere quando essa ha un valore pari a zero; quindi, quando più di n vincoli sono attivi nello stesso punto dello spazio.

2. Enuncia i metodi per partire da una soluzione di base ammissibile.

I metodi per partire da una soluzione di base ammissibile sono tre:

1. Metodo delle variabili artificiali: vengono introdotte delle variabili artificiali all'interno del problema di programmazione lineare per andare a creare un problema artificiale. Questo nuovo problema è utile in quanto è ammissibile per costruzione, permettendoci cosí di applicare l'algoritmo del simplesso su di esso. Una volta trovata la soluzione ottima, se essa vale zero allora si è trovata una soluzione ammissibile per il problema, altrimenti il problema originario è inammissibile.
2. Metodo "big M": a differenza della soluzione precedente, questa tecnica non elimina dalla formulazione le variabili x, ma penalizza nell'obiettivo le variabili u con coefficienti molto grandi. In questo modo qualsiasi soluzione con una variabile u maggiore di zero avrà un costo maggiore del valore ottimo del problema e quindi vengono penalizzate.
3. Metodo di Balisky-Gomory: viene minimizzata una funzione obiettivo ausiliaria la quale misura l'inammissibilità del problema rispetto ad un vincolo violato. Questo procedimento viene eseguito fino a quando non si arriva ad una soluzione ammissibile o si dimostra l'inammissibilità. In questa situazione può capitare che il problema ausiliario sia illimitato, se cosí fosse si può prendere come pivot un valore negativo e provare ad applicare l'algoritmo del simplesso.

3. Come si valuta la robustezza di una soluzione?

La robustezza di una soluzione può essere valutata in due modi:

1. Analisi di sensitività: questa tipologia di analisi va a valutare l'intervallo nel quale può variare ogni coefficiente cj e bj senza andare a cambiare la base ottima.
2. Analisi parametrica: questa tipologia di analisi studia come la soluzione ottima dipende dal valore del termine noto di un determinato vincolo. Il risultato sarà una funzione lineare a tratti dove ogni suo segmento corrisponde ad una base ottima ed ogni punto di discontinuità ad un cambio di base.

4. Cosa sono i prezzi ombra?

I prezzi ombra sono i coefficienti di costo ridotto delle colonne di slack all'ottimo, cioè i prezzi massimi a cui conviene comprare la risorsa e i prezzi minimi a cui conviene venderla. Nel caso di risorse non scarse i prezzi ombra sono nulli.

5. Programmazione multilivello.

La programmazione multilivello o multi-obiettivo è una situazione in cui il decisore deve gestire molti obiettivi. Essi sono in conflitto tra loro quando migliorando uno vengono peggiorati gli altri. Il processo di risoluzione di questa tipologia di problemi si divide in due fasi distinte:

1. Trovare la regione Pareto-ottima: insieme delle soluzioni ammissibili non dominate.
2. Scelta di una soluzione tra quelle individuate nella prima fase.

6. Dare la definizione di dominanza nella programmazione a molti obiettivi.

Date due soluzioni x' e x'', x' domina la soluzione x'' se e solo se:

1. Fi(x') <= Fi(x'') per ogni i.
2. Esiste una j tale che Fj(x') < Fj(x'').

7. Come si identifica una regione Pareto-ottima?

Una regione di Pareto-ottima è un insieme di soluzioni ammissibili non dominate, dove date due soluzioni x' e x'', x' domina la soluzione x'' se e solo se:

3. Fi(x') <= Fi(x'') per ogni i.
4. Esiste una j tale che Fj(x') < Fj(x'').

Questa regione di Pareto-ottima viene identificata attraverso due tecniche:

1. Metodo dei pesi: ottimizza una combinazione convessa delle funzioni obiettivo pesandole con un valore y. Quindi rappresenta tutte le soluzioni che sono ottime per un certo valore di y.
2. Metodo dei vincoli: ottimizza una funzione obiettivo, trasformando le altre in vincoli introducendo un termine noto parametrico.

8. Come viene scelta la soluzione ottima nella programmazione a molti obiettivi?

Ci sono diversi metodi con cui viene selezionata la soluzione ottima dopo aver identificato la regione di Pareto-ottima:

1. Metodo delle curve di indifferenza: la soluzione scelta è quella in cui la curva di indifferenza è tangente alla regione Paretiana.
2. Criterio del punto di massima curvatura: la soluzione scelta è quella per cui un piccolo miglioramento di un obiettivo corrisponde ad un grande peggioramento dell'altro.
3. Criterio del punto-utopia: la soluzione scelta è quella che ha come coordinate, nello spazio degli obiettivi, i valori ottimi di ciascuna funzione obiettivo.
4. Criterio degli standard: la soluzione scelta è quella che ha come coordinate, nello spazio degli obiettivi, i valori standard di ciascuna funzione obiettivo. I valori standard sono i valori sotto alla quale non si vuole che gli obiettivi peggiorino.

9. Enunciare il teorema della dualità nella forma debole e nella forma forte.

Il teorema della dualità in forma debole ci dice che dati due problemi primale e duale possiamo dire che per ogni soluzione ammissibile del primale e per ogni soluzione ammissibile del duale abbiamo che ct*x(ammissibile) <= bt*y(ammissibile). Il teorema della dualità in forma forte invece, ci dice che dati due problemi primale e duale possiamo dire che se il primale ha un valore ottimo esso sarà congruente a ct*x(ottima), il che è uguale a bt*y(ottima) e quindi congruente al valore ottimo del duale.

10. Lemma di Farkas, enunciarlo, a cosa serve, dove si usa.

Il lemma di Farkas dice che dato un sistema di equazioni lineari del tipo Ax = b per x >= 0, una e una sola di queste alternative è vera:

1. Esiste una soluzione ammissibile per il primale.
2. Esiste una soluzione ammissibile per il duale con le condizioni At * y >= 0, bt * y < 0.

Esso ci permette di dedurre che data una coppia primale e duale, non è possibile che entrambi abbiamo una soluzione ammissibile. Questo lemma viene usato nella programmazione lineare per problemi che sfruttano equazioni lineari.

11. Enunciare il teorema dello scarto complementare.

Il teorema dello scarto complementare dice che data una coppia di problemi primale-duale, una condizione necessaria e sufficiente per l'ottimalitá delle due soluzioni ammissibili x e y è che valgano le seguenti equazioni:
y(ammissibile)t * (b – A * x(ammissibile)) = 0.
(At * y(ammissibile) - c) * x(ammissibile) = 0.

12. Simplesso duale come funziona e perché lo usiamo?

L'algoritmo del simplesso duale è una variante dell'algoritmo del simplesso, il quale ci permette di lavorare sul tableau del primale per perseguire l'ammissibilità conservando l'ottimalitá. Questo viene fatto sfruttando un nuovo metodo di scelta del pivot:

1. Viene scelta una riga con termine negativo.
2. Si seleziona poi un pivot negativo che minimizzi il valore assoluto del rapporto tra quest'ultimo e il valore del coefficiente di costo ridotto.

Questo algoritmo viene utilizzato nel caso in cui il problema ha una base iniziale inammissibile per il primale e super-ottima per il duale.

13. Descrivere quali sono i criteri per eseguire un passo di pivot nell'algoritmo del simplesso duale.

L'algoritmo del simplesso duale è una variante dell'algoritmo del simplesso, il quale ci permette di lavorare sul tableau del primale per perseguire l'ammissibilità conservando l'ottimalitá. Questo viene fatto sfruttando un nuovo metodo di scelta del pivot:

1. Viene scelta una riga con termine negativo.
2. Si seleziona poi un pivot negativo che minimizzi il valore assoluto del rapporto tra quest'ultimo e il valore del coefficiente di costo ridotto.

14. Bound duale / Bound primario

Nel caso della programmazione lineare intera per definire l'ottimalitá si calcola un upper bound e lower bound tali che la soluzione ottima sia compresa tra di loro. Nel caso di un problema di minimizzazione il bound primale è quello superiore, mentre quello duale è quello inferiore (il contrario nel caso di massimizzazione). Quindi un bound primale è dato dal valore della funzione obiettivo z(x) in una qualsiasi soluzione ammissibile x e può essere calcolato attraverso: algoritmi euristici o meta-euristici e con algoritmi di approssimazione con garanzia. Un bound duale invece, è dato dal valore della funzione obiettivo z(x) in una qualsiasi soluzione super-ottima, la quale viene calcolata risolvendo l'ottimo di un rilassamento o trovando una soluzione ammissibile y del duale.

15. Dare la definizione di rilassamento.

Il rilassamento R di un problema P viene definito come R = min{Zr(x) : x e X} se:

1. La regione ammissibile di P è inclusa in quella di R.
2. La soluzione ottima di R è migliore o uguale rispetto a quella di P.

Esistono 4 diverse tipologie di rilassamenti che ci permettono di ottenere i bound. Esse sono:

1. Rilassamento lineare.
2. Rilassamento combinatorio.
3. Rilassamento lagrangiano.
4. Rilassamento surrogato.

Noi ci concentreremo durante il corso solo sul rilassamento nel continuo, dove i problemi di ottimizzazione hanno una formulazione unica. Nel discreto questi problemi non hanno una formulazione unica e il nostro obiettivo è andare ad identificare quella ideale che ci consente di risolverlo come se fosse un problema di programmazione lineare nel continuo. Questa formulazione

ottimale puó essere definita con un poliedro che corrisponde al guscio convesso delle soluzioni interne. La disciplina che si occupa di selezionare e migliorare le formulazioni lineari di problemi di programmazione intera è della Polyhedral Combinatorics.
Una delle modaltá principali con il quale si riesce a restringere la formulazione di problemi di ottimizzazione discreta è quello di disgregare i vincoli .

16. Cosa sono e a cosa servono gli algoritmi "cutting planes"?

Gli algoritmi "cutting planes" sono una tipologia di algoritmi che ci permettono di rendere più stringenti le formulazioni del problema di programmazione discreta, in modo tale da avvicinarci al guscio convesso delle soluzioni interne, ovvero la formulazione ideale. Quindi alla fine avremo una formulazione più stretta del problema, data dall'aggiunta del vincolo Qx <= q, dove valgono le condizioni: Qx <= q per tutte le x appartenenti a Zn e che Qx* > q, ovvero che la soluzione ottima del problema nel discreto deve essere inammissibile, permettendo cosí di escluderla.

17. Taglio di Gomory, come si fa a generare?

Il taglio di Gomory è una tecnica particolare che ci permette di trovare facilmente disuguaglianze valide sfruttando la procedura di Chavatal-Gomory. Esso è possibile trovarlo andando a prendere un problema di programmazione lineare discreto P e il suo rilassamento LP (ignora le condizioni di integralità). Dopo di che viene calcolata x* e z*, ovvero la soluzione ottime di LP, avente come valore la costante (presente in alto a sinistra nel tableau) sommata con le colonne fuori base alla riga 0. Come vincoli viene posto che: la soluzione x* deve essere maggiore o uguale a zero e che l'insieme delle variabili di base in riga i sommate con la sommatoria delle variabili non in base debba essere uguale al termine noto del tableau alla colonna 0. Una volta calcolata la soluzione ottima viene sfruttata la procedura di Chavatal-Gomory sulla riga i dove x* non ha un valore intero, arrotondando per difetto il termine noto, e sottratta al vincolo di uguaglianza sopra citato. Quello che si ottiene da questa sottrazione è proprio il taglio di Gomory.

18. Branch and bound. Abbandono l'ottimo per una soluzione approssimata in modo da avere un algoritmo che non ci metta troppo, come faccio?

L'algoritmo branch and bound è utile ad eliminare alcune soluzioni per non farle crescere in maniera esponenziale con l'aumentare del numero di variabili del problema di programmazione discreta. Esso si divide in due fasi:

1. La fase di branching, la quale ci permette di dividere il nostro problema di programmazione discreta in diversi sotto-problemi (a seconda che si tratti di un branching binario e n-ario), mantenendo la correttezza, quindi senza avere perdite di soluzioni, e l'efficienza, avendo quindi tutti sottoinsiemi disgiunti tra loro. Esistono due diversi modi per fare branching, tramite il fissaggio di variabili (una variabile binaria, una variabile intera o n variabili binarie) o tramite l'inserzione di vincoli (di vincoli interi composti dalle variabili x1..xn, dal vettore di coefficienti interi a1..an e da un termine noto k, permettendoci cosí di avere due sotto-problemi ax <= k e ax >= k + 1. Questa fase termina quando troviamo un sotto-problema inammissibile, risolto all'ottimo o trascurabile (questo sì può determinare risolvendo il rilassamento del sotto-problema).
2. La fase di bounding, la quale ci permette di associare un bound duale ad ogni sotto-problema per essere poi confrontato con il bound primale. Nel caso in cui il bound duale del rilassamento del sotto-problema considerato sia maggiore o uguale del bound primale (quindi peggiore) possiamo scartarlo. Per garantire la correttezza della fase di bounding andiamo a concatenare due disequazioni: la prima dice che non può esistere una soluzione

di X(F) con un valore migliore del suo rilassamento, mentre la seconda che non esiste una soluzione del rilassamento del sotto-problema con valore migliore della soluzione ammissibile del problema originale.

Questo ci fa capire che è inutile risolvere il problema F all'ottimo perché non può fornire soluzione migliore di quella ammissibile per P. Questo algoritmo è fondamentale per risparmiare tempo e memoria.

19. Dare la definizione di minimo locale.

Quando si devono studiare problemi non lineari, si deve introdurre il concetto di ottimalitá locale e ottimalitá globale. Nel caso dell'ottimalitá locale essa viene definita da una soluzione x che viene detto minimo locale se per ogni e > 0 la funzione del minimo locale è minore o uguale rispetto a ogni punto x appartenente a X, tale che il valore assoluto della differenza tra il minimo locale e questa soluzione x scelta sia minore o uguale di e.

20. Cos'è la programmazione convessa?

La programmazione convessa è una particolare eccezione che ci permette di identificare l'ottimo globale, cosa difficile da fare in quanto si dovrebbero andare ad enumerare tutti gli ottimi locali. Un problema di minimizzazione non lineare è convesso quando:

1. La funzione obiettivo è una funzione convessa, ovvero che per ogni coppia di punti x1 e x2 e per 0 <= L <= 1 allora f(Lx1 + (1-L)x2) <= Lf(x1) + (1-L)f(x2).
2. La regione ammissibile è un insieme convesso, ovvero che per ogni coppia di punti x1 e x2 e per 0 <= L <=1 allora Lx1 + (1-L)x2 appartiene a X. Quindi è convessa quando tutti i vincoli di uguaglianza sono lineari e quando tutti i vincoli di disuguaglianza sono convessi.

21. Cosa sono gli algoritmi iterativi?

Gli algoritmi per l'ottimizzazione dei problemi d'ottimizzazione non lineare vengono detti algoritmi iterativi, i quali convergono verso un minimo locale. Questi algoritmi partono da una soluzione data x(0) e calcolano la sequenza di soluzioni tali che il valore di f(x) diminuisca mono-tonicamente. Essi si fermano quando il miglioramento ottenuto o il passo compiuto sono più piccoli di una data soglia.
Ad ogni iterazione l'algoritmo calcola una direzione d(k) e un passo s(k).
Essi devono avere delle caratteristiche fondamentali:

1. Convergenza globale: l'algoritmo converge ad un ottimo locale ma lo fa globalmente, ovvero partendo da un qualunque punto iniziale.
2. Dipende dalla velocità con cui converge.

Le due principali strategie per costruire algoritmi iterativi sono:

1. Algoritmi line search, i quali si concentrato a scegliere prima la direzione e poi il passo.
2. Algoritmi trust region, i quali trovano prima il passo e poi vanno a scegliere la direzione.

22. Dare la definizione di velocità di convergenza.

La velocità di convergenza definisce la rapidità con cui un algoritmo converge al minimo locale. Essa cambia a seconda che abbiamo una:

1. Convergenza lineare.
2. Convergenza super-lineare
3. Convergenza quadratica.

23. Descrivere l'algoritmo del gradiente.

L'algoritmo del gradiente è una delle tecniche che viene utilizzata per scegliere la direzione che dovrà prendere l'algoritmo iterativo per trovare l'ottimalitá del problema di programmazione non lineare. L'algoritmo va a calcolare il valore della funzione f valutata nel passo successivo, dato dalla somma del punto attuale e il prodotto tra direzione e passo, applicando la successione di Taylor fino al fattore di primo ordine. A questo punto si può notare che a far decrescere più rapidamente la funzione f è proprio quella opposta a quella del gradiente.

24. Metodo di Newton.

L'algoritmo di Newton è una delle tecniche che viene utilizzata per scegliere la direzione che dovrà prendere l'algoritmo iterativo per trovare l'ottimalitá del problema di programmazione non lineare. L'algoritmo, come visto in quello del gradiente, va a calcolare il valore della funzione f valutata nel passo successivo, dato dalla somma del punto attuale e il prodotto tra direzione e passo, applicando la successione di Taylor fino al fattore di secondo ordine. A questo punto si può notare che a far decrescere più rapidamente la funzione f è proprio la direzione definita da quella del gradiente moltiplicata per il negato dell'inverso di una matrice B semi-definita positiva.

25. Nella PNL, quando ho scelto la direzione, come scelgo il passo?

Nella programmazione non lineare una volta scelta la direzione mediante una delle due tecniche, ovvero quella del gradiente o quella di Newton, viene scelto il passo. Questa operazione può essere fatta mediante due algoritmi:

1. Metodo della bisezione, il quale va a sfruttare il calcolo della derivata.
2. Metodo dei numeri di Fibonacci, il quale viene definito derivate free, in quanto non sfrutta il calcolo della derivata.

26. Metodo della bisezione.

Il metodo della bisezione è uno delle tecniche che vengono utilizzate per determinare il passo mediante il calcolo della derivata. Abbiamo un intervallo [a,b] in cui può provarsi il nostro punto di minimo, andiamo quindi a calcolare la deriva nel punto centrale dell'intervallo. Se questo valore risulta essere positivo, allora restringiamo l'intervallo in [a, (a+b)/2] e significa che la funzione sale. Se il valore risulta essere negativo, allora l'intervallo da considerare sarà [(a+b)/2, b] e significa che la funzione scende. Si ripete questo procedimento fino a quando l'intervallo non risulta essere abbastanza piccolo.

27. Cosa dice la condizione di Karush Kuhn Tucker?

La condizione di Karush Kuhn Tucker permette di stabilire delle condizioni di primo ordine, che devono essere rispettate affinché un punto nella funzione Lagrangiana sia un minimo locale. Le condizioni sono:

1. Il gradiente di x* rispetto alla funzione Lagrangiana deve essere zero.
2. Tutti i vincoli di uguaglianza devono essere soddisfatti.
3. Tutti i vincoli di disuguaglianza devono essere soddisfatti.
4. Lambda i devono essere non negativi per i vincoli di disuguaglianza.
5. Deve valere la condizione di complementarità, ovvero almeno uno dei due tra il vincolo e lambda deve essere zero.

29. Algoritmo del Simplesso rivisto Simplesso rivisto - ha disegnato la matrice, le formule e descritto i passi [per il 30L].