RENDICONTI
del
SEMINARIO MATEMATICO
della
UNIVERSITÀ DI PADOVA

G. ANDREATTA

Problemi di ottimizzazione nella teoria delle code

Rendiconti del Seminario Matematico della Università di Padova, tome 53 (1975), p. 123-134

<http://www.numdam.org/item?id=RSMUP_1975__53__123_0>

REND. SEM. MAT. UNIV. PADOVA, Vol. 53 (1975)

Problemi di ottimizzazione nella teoria delle code.

G. ANDREATTA (*)

Introduzione.

Nel 1957 Bellman (cfr. [2], cap. XI) propose di arricchire la nozione di catena (finita ed omogenea) di Markov aggiungendovi:

a) delle « ricompense » associate ad ogni transizione effettuata dal processo in evoluzione stocastica;

b) la possibilità di influire sul processo prendendo certe « decisioni » atte a modificare le probabilità e le ricompense delle successive transizioni.

Su un modello stocastico così arricchito (denominato catena di Markov con decisione) è allora naturale cercare di instaurare un procedimento di « ottimizzazione »: questi suggerimenti di Bellman sono stati ripresi nel 1960 da Howard (cfr. [4]) il quale ha messo a punto un algoritmo iterativo capace di determinare, sotto opportune ipotesi, le « decisioni ottime ».

In questo lavoro prenderemo in esame quella particolare classe di catene di Markov con decisione costituita dai modelli M/M della Teoria delle Code: a questi processi non è però applicabile l'algoritmo di Howard, poichè, come faremo vedere in un successivo paragrafo, una delle ipotesi richieste da Howard [4] non è verificata, salvo casi particolari.

(*) Indirizzo dell'A.: Istituto di Statistica dell'Università di Padova.

Lavoro eseguito nell'ambito dell'attività dei gruppi di Ricerca matematica del C.N.R.

Finora questo problema è stato parzialmente risolto, nel caso in cui la decisione da prendere sia il numero delle « stazioni di servizio » da attivare determinando una volta per tutte questo numero, indipendentemente dal numero di « clienti » in coda (lo stato del processo): a questo modo però si esclude la possibilità di un dimensionamento dinamico (in cui cioè il numero di stazioni di servizio possa variare in dipendenza del numero di clienti in coda).

In questo lavoro si affronta e risolve questo problema, mediante l'esposizione di un nuovo algoritmo iterativo che appartiene sostanzialmente alla Programmazione Dinamica.

1. Notazioni. Terminologia. Ipotesi.

Sia $n+1$ il numero degli stati accessibili al processo: indicheremo tali stati coi simboli $E_0, \ldots, E_n$.

Quando il processo si trova nello stato E_i si deve scegliere fra un certo numero (finito) di « alternative »: sia U_i l'insieme di tali alternative (relative allo stato E_i) e u_i un suo generico elemento.

Chiameremo « strategia » e la indicheremo con u (oppure w) qualunque regola che ad ogni stato associ in qualche modo una alternativa: una strategia quindi è una famiglia di alternative $u = (u_0, \ldots, u_n)$; sia U la totalità di tutte le strategie: è evidente che U è il prodotto cartesiano degli insiemi U_i $(i = 0, \ldots, n)$.

Poichè il processo evolve in dipendenza di un certo parametro (che spesso ha fisicamente il significato del tempo) è naturale associare ad ogni valore del parametro una strategia (che a priori non è detto sia sempre la stessa): alla famiglia di strategia che ne risulta daremo il nome di « politica ».

Ottimizzare un processo vorrà dire per noi trovare una politica a cui corrisponda il massimo della ricompensa media per unità di « tempo » in regime ergodico, nell'ipotesi che il processo continui la sua evoluzione per un tempo che si prolunga all'infinito.

Blackwell ha dimostrato (cfr. [3]) che fra le politiche ottimali ve n'è una stazionaria, che associa cioè ad ogni valore del parametro la medesima strategia (daremo l'appellativo di « ottima » ad una tale strategia): di conseguenza per ottimizzare un processo è sufficiente trovare una strategia ottima.

Indicheremo con $g(u)$ la ricompensa media per unità di tempo in

regime ergodico (in inglese « gain »: cfr. [4]) relativa alla politica stazionaria ottenuta a partire dalla strategia u.

Per le rimanenti notazioni e terminologie faremo in generale riferimento a quelle usate da Howard in [4].

Howard nel suo lavoro ipotizza che la ricompensa associata ad una effettiva transizione, ricompensa che indica col simbolo r_{ij}, dipenda solo dalla alternativa u_i relativa allo stato « iniziale » della transizione mentre è ragionevole supporre che in generale tale ricompensa dipenda anche dall'alternativa u_j relativa allo stato « finale » come prova il seguente esempio: pensiamo ad un processo di code concreto in cui la decisione da prendere sia il numero di stazioni di servizio da aprire, tenuto conto che ogni stazione di servizio funzionante comporta dei costi e che si vuole minimizzare il costo medio di esercizio: se improvvisamente, perchè è aumentato il numero di clienti in coda (lo stato), si decide di aumentare il numero di stazioni di servizio (l'alternativa) si dovrà probabilmente sostenere un costo impulsivo (una ricompensa negativa) dovuto all'aumento del numero di stazioni: tale costo proprio perchè legato ad un aumento e cioè ad una variazione dipende da due successive alternative (quella « iniziale » e quella « finale »).

A differenza di Howard noi ci poniamo nel caso più generale: indicheremo con $r_{ij}(u_i, u_j)$ la generica ricompensa.

Supporremo invece verificate (come in [4]) le seguenti due ipotesi:

H1) ad ogni strategia sia associato un processo completamente ergodico;

H2) il numero delle alternative relative ad uno stato qualunque sia finito.

2. L'algoritmo di risoluzione: descrizione generale.

L'algoritmo di risoluzione è di tipo iterativo e ciascuna iterazione è costituita di due « fasi » che, in analogia alla nomenclatura di Howard (VDO = Value-Determination Operation, PIR = Policy-Improvement Routine), denomineremo rispettivamente « SIR » e « GDO » (Strategy-Improvement Routine e Gain-Determination Operation).

Prima di descrivere queste due fasi, introduciamo per ogni fissata strategia w una funzione $\Delta_w(u)$ da U in R a cui daremo il nome di « pseudo-scarto ». Supposto $\pi_0(u)$ (la probabilità asintotica che il processo si trovi nello stato E_0 (cfr. [3]) diverso da zero qualunque sia la

strategia u [1], sia:

$$\Delta_w(u) = \frac{1}{\pi_0(u)} \cdot (g(u) - g(w)) \,. \tag{2.1}$$

Sono evidenti le seguenti proporzioni:

P1) la funzione pseudo-scarto $\Delta_w(u)$ è maggiore (rispettivamente: uguale, minore) di zero se e soltanto se la ricompensa media $g(u)$ relativa alla strategia u è maggiore (rispettivamente: uguale, minore) della ricompensa media $g(w)$ relativa alla strategia w;

P2) fissata w, il massimo (al variare di u in U) [2] della funzione pseudo-scarto $\Delta_w(u)$ è maggiore o uguale a zero;

P3) se, fissata w, il massimo di $\Delta_w(u)$ è uguale a zero allora w è politica ottima;

P4) se, fissata w, il massimo di $\Delta_w(u)$ è maggiore di zero e u^* è una strategia in cui $\Delta_w(u)$ assume tale valore massimo allora la ricompensa media associata alla strategia u^* è maggiore della ricompensa media associata alla strategia w [3].

Nella fase SIR si calcola appunto il massimo (al variare di u) della funzione pseudo-scarto $\Delta_w(u)$: le proposizioni P2, P3, P4 dimostrano che a partire da una strategia w la SIR permette di stabilire se w è una strategia ottima (nel qual caso il procedimento si arresta) altrimenti fornisce una strategia u^* « migliore ».

Nella fase GDO, per ogni strategia u^* trovata dalla SIR si calcola la corrispondente ricompensa media $g(u^*)$ mediante la seguente formula:

$$g(u^*) = \pi_0(u^*) \cdot \Delta_w(u^*) + g(w) \tag{2.2}$$

dopo di che si inizia una nuova iterazione sostituendo u^* a w nella fase SIR.

Il procedimento si innesca nella fase SIR fingendo che esista una strategia w a cui corrisponde una ricompensa media nulla $(g(w) = 0)$ [4].

---

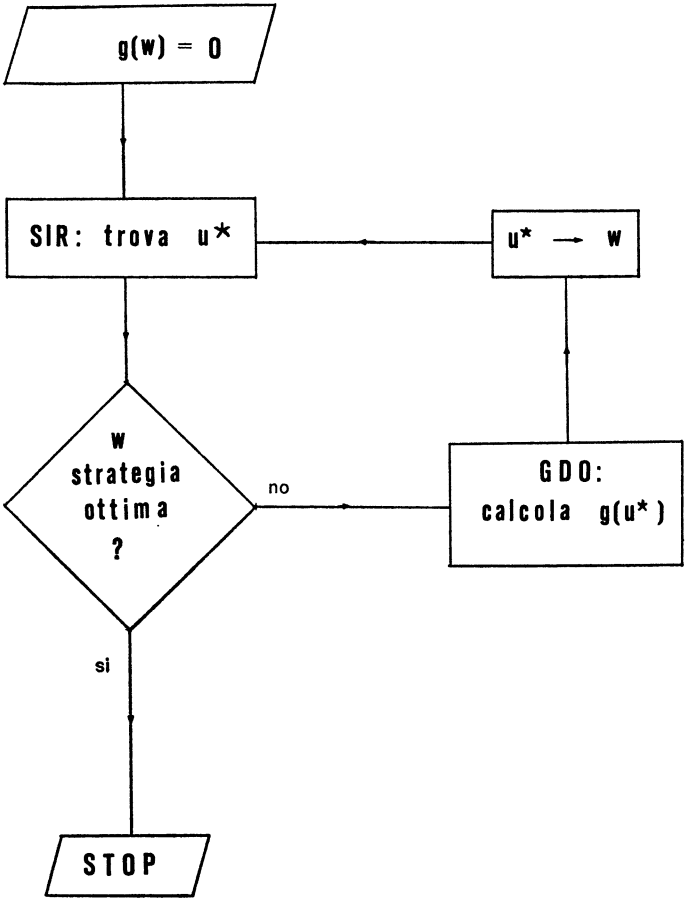
[1] Il caso generale comporta complicazioni unicamente di carattere formale.

[2] L'esistenza di tale massimo è garantita dall'ipotesi H2.

[3] Naturalmente non è detto ce u^* sia una strategia ottima; per esserlo dovrebbe massimizzare anche la funzione $g(u)$.

[4] Supponendo che le ricompense siano non negative; in caso contrario si pone $g(w)$ uguale ad un numero negativo opportunamente scelto.

L'algoritmo può quindi essere schematizzato nel modo seguente:



OSSERVAZIONI:

1) la convergenza dell'algoritmo è assicurata dall'ipotesi H2 e dalle proposizioni P2, P3, P4;

2) i risultati numerici hanno dimostrato una discreta rapidità di convergenza: per esempio in problemi con una decina di stati e una decina di alternative per ogni stato (il numero delle strategie possibili è quindi dell'ordine di 10^{10}) si è avuta la convergenza entro la sesta iterazione;

3) calcolare il massimo della funzione pseudo-scarto $\Delta_w(u)$ nel caso di un processo non di code è in generale tutt'altro che facile e spesso materialmente impossibile dati i limiti dei mezzi di calcolo a disposizione: per tale motivo abbiamo ristretto le nostre considerazioni proprio ai processi di code per i quali tale calcolo risulta sicuramente possibile, come faremo vedere;

4) per calcolare $g(u^*)$ si potrebbe utilizzare anche la VDO di Howard [4]: è però più conveniente utilizzare la GDO perchè a differenza della VDO non occorre risolvere un sistema lineare: per calcolare $\pi_0(u^*)$ si sfrutta la formula (3.3) del successivo paragrafo.

3. Teoria delle Code: alcuni richiami.

Consideriamo quei processi di code che sono markoviani, omogenei e finiti (cioè i modelli M/M chiusi).

Per tali processi le « velocità di transizione » (in inglese « transition rates ») a_{ij} sono uguali a zero non appena $|i-j| \geqslant 2$; nella letteratura si usano spesso i simboli λ_i anzichè $a_{i,i+1}$ $(i = 0, \ldots, n-1)$ e μ_i anzichè $a_{i,i-1}$ $(i = 1, \ldots, n)$. Naturalmente λ_i e μ_i dipendono dalla strategia u e più precisamente dalla sua componente i-esima: la alternativa u_i.

Per rendere meno pesante la trattazione assumeremo la seguente ipotesi (peraltro non restrittiva [5]):

C1) $$\lambda_i(u_i) \neq 0 \qquad \forall u_i \in U_i \qquad (i = 0, \ldots, n-1)$$

$$\mu_i(u_i) \neq 0 \qquad \forall u_i \in U_i \qquad (i = 1, \ldots, n)$$

Si osservi che l'ipotesi C1 implica che gli stati $E_0, \ldots, E_n$ costituiscono una unica catena irriducibile: risulta perciò soddisfatta l'ipotesi H1 (il processo risulta completamente ergodico), inoltre, non essendoci stati transienti, $\pi_0(u)$ risulta diverso da zero qualunque sia la strategia u di U (ipotesi formulata nella definizione della funzione pseudo-scarto $\Delta_w(u)$).

---

[5] Se per esempio $\mu_i = 0$ tutti gli stati $E_0, \ldots, E_{i-1}$ risultano transienti e perciò non incidono sulla ricompensa media in regime ergodico: possiamo quindi rinumerare gli stati dopo di che risulta verificata l'ipotesi C1.

Dalla Teoria delle Code è noto che (nell'ipotesi che $\mu_k(u_k) \neq 0$):

$$\pi_k(u) = \pi_{k-1}(u) \cdot \frac{\lambda_{k-1}(u_{k-1})}{\mu_k(u_k)} \qquad (k = 1, \ldots, n) \text{ (6)} \tag{3.1}$$

da cui consegue:

$$\frac{\pi_k(u)}{\pi_0(u)} = \begin{cases} \prod_{s=1}^{k} \frac{\lambda_{s-1}(u_{s-1})}{\mu_s(u_s)} & k = 1, \ldots, n\,, \\ 1 & k = 0\,, \end{cases} \tag{3.2}$$

che a sua volta implica:

$$\pi_0(u) = \left[1 + \frac{\lambda_0(u_0)}{\mu_1(u_1)} \cdot \left(1 + \frac{\lambda_1(u_1)}{\mu_2(u_2)} \cdot \left(\ldots \left(1 + \frac{\lambda_{n-1}(u_{n-1})}{\mu_n(u_n)}\right) \ldots \right)\right)\right]^{-1}. \tag{3.3}$$

Si noti che mentre ciascun $\pi_k(u)$ è funzione di tutte le alternative $(u_0, \ldots, u_n)$, $\pi_k(u)/\pi_0(u)$ dipende solo dalle prime $k+1$ alternative $(u_0, \ldots, u_k)$.

Ricordiamo che in [4] si dimostra che la ricompensa media si può ricavare dalla seguente formula:

$$g(u) = \sum_{i=0}^{n} \pi_i(u) \cdot q_i(u) \tag{3.4}$$

dove nel caso generale $q_i(u)$ è dato dalla seguente espressione:

$$q_i(u) = r_{ii}(u_i) + \sum_{j \neq i} a_{ij}(u_i) \cdot r_{ij}(u_i, u_j) \tag{3.5}$$

che nel caso di un modello M/M chiuso (con la convenzione $\lambda_n = \mu_0 = 0$) diventa:

$$q_i(u) = \mu_i(u_i) \cdot r_{i,i-1}(u_i, u_{i-1}) + r_{ii}(u_i) + \lambda_i(u_i) \cdot r_{i,i+1}(u_i, u_{i+1})\,. \tag{3.6}$$

4. Modello M/M chiuso: computabilità della SIR.

Fissata una strategia w, poniamo:

$$F_k(u_{k-1}) = \operatorname{Max}\left[\sum_{i=k}^{n} \left[\frac{\pi_i(u)}{\pi_{k-1}(u)} \cdot (q_i(u) - g(w))\right] + \lambda_{k-1}(u_{k-1}) \cdot r_{k-1,k}(u_{k-1}, u_k)\right]$$

(6) Ricordiamo che π_k è la probabilità asintotica che il processo si trovi nello stato E_k.

al variare di $u_k, \ldots, u_n$ in modo che $(u_k, \ldots, u_n) \in \mathfrak{U}_k \times \ldots \times \mathfrak{U}_n$ se $k > 0$, e

$$F_0 = \operatorname*{Max}_{u \in \mathfrak{U}} \sum_{i=0}^{n} \frac{\pi_i(u)}{\pi_0(u)} \cdot (q_i(u) - g(w)) \, .$$

Dalla (3.4) e per ben note proprietà probabilistiche è evidente che si ha:

$$F_0 = \operatorname*{Max}_{u \in \mathfrak{U}} \Delta_w(u) \, . \tag{4.1}$$

L'importanza delle funzioni $F_k(u_{k-1})$ ed F_0 risiede nel fatto che essendo (come dimostreremo) funzioni di tipo ricorsivo permettono di atteggiare a problema dinamico a stadi successivi la ricerca del massimo della funzione pseudo-scarto. Infatti:

TEOREMA. Sussistono le seguenti relazioni di tipo ricorsivo:

$$F_k(u_{k-1}) = \operatorname*{Max}_{u_k \in \mathfrak{U}_k} \left\{ [F_{k+1}(u_k) + r_{kk}(u_k) + \mu_k(u_k) \cdot r_{k,k-1}(u_k, u_{k-1}) - g(w)] \cdot \right.$$
$$\left. \cdot \frac{\lambda_{k-1}(u_{k-1})}{\mu_k(u_k)} + \lambda_{k-1}(u_{k-1}) \cdot r_{k-1,k}(u_{k-1}, u_k) \right\} \qquad k = 1, \ldots, n \, ,$$

$$F_0 = \operatorname*{Max}_{u_0 \in \mathfrak{U}_0} [F_1(u_0) + r_{00}(u_0) - g(w)] \, .$$

DIMOSTRAZIONE. Siamo infatti, tenuto conto delle (3.1), (3.6) e dopo opportune identificazioni di simboli, nelle ipotesi del corollario 1.6.2 di [5], cap. 2°, il quale, fra l'altro, garantisce l'asserto.

Si osservi che il « Principio di Ottimalità » di Bellman (cfr. [2]) afferma quello che il Teorema dimostra (7).

Il calcolo di $F_n(u_{n-1})$ mostra come dobbiamo scegliere u_n in funzione di u_{n-1}: indichiamo tale alternativa con $u_n^*(u_{n-1})$; analogamente il calcolo di $F_i(u_{i-1})$, quando $i > 0$, mostra come scegliere u_i in funzione di u_{i-1}: indichiamola con $u_n^*(u_{i-1})$ e finalmente il calcolo di F_0 porge l'alternativa migliore u_0^*.

Una strategia che sicuramente massimizza la funzione pseudo-scarto è quindi la strategia composta dalle seguenti alternative:

$$u_0 = u_0^*$$
$$\ldots \ldots$$
$$u_i = u_i^*(u_{i-1}^*)$$
$$\ldots \ldots$$
$$u_n = u_n^*(u_{n-1}^*) \, .$$

(7) Sulla problematica che nasce dal « principio » di ottimalità si veda Volpato ([5], cap. 2).

Con questo Teorema e le considerazioni appena svolte è provata la effettiva computabilità della SIR (e quindi dell'intero algoritmo) nel caso di processi della Teoria delle Code.

5. Un procedimento di subottimizzazione.

Agli effetti operativi è spesso sufficiente trovare una strategia la cui ricompensa media sia vicina (con una approssimazione arbitrariamente prefissata) a quella ottima.

In questo paragrafo descriveremo una variante dell'algoritmo visto finora che consentirà di individuare una strategia $\hat{u}$ arbitrariamente « vicina » ([8]) ad una strategia ottima con il vantaggio che mentre il numero di iterazioni richiesto dall'algoritmo originario (pur essendo sicuramente finito) non è noto a priori e può magari risultare piuttosto elevato, quello invece richiesto dalla variante che descriveremo ora possiede un confine superiore noto a priori (in funzione dell'approssimazione richiesta).

Ricordiamo che la GDO fornisce alla SIR un numero (che è la ricompensa media relativa ad una certa strategia w), dopo di che la SIR permette di stabilire se esiste oppure no una strategia a cui corrisponda una ricompensa media il cui valore sia superiore a quel numero.

Che cosa succede se alla SIR forniamo un numero a caso c (senza che esso rappresenti necessariamente la ricompensa di qualche strategia)?

Riscriviamo la funzione pseudo-scarto come segue:

$$\Delta_c(u) = \frac{1}{\pi_0(u)} \cdot (g(u) - c) \,. \tag{5.1}$$

Risultano evidenti le seguenti proposizioni (conveniamo, per evitare inutili ripetizioni, che u^* sia una strategia ottima e $\hat{u}$ una strategia tale che $\Delta_c(\hat{u}) = \operatorname*{Max}_{u \in \mathfrak{U}} \Delta_c(u)$):

P1′) la funzione pseudo-scarto $\Delta_c(u)$ è maggiore (rispettivamente: uguale, minore) di zero se e soltanto se il valore della ricompensa media $g(u)$ relativa alla strategia u è maggiore (rispettivamente: uguale, minore) del numero c;

([8]) Tale cioè che $|g(\hat{u}) - g(u^*)| < \varepsilon$ (se ε è un prefissato numero positivo e u^* una strategia ottima).

P2′) fissato c, il massimo di $\Delta_c(u)$ può risultare sia maggiore sia uguale sia minore di zero;

P3′) se $\operatorname{Max} \Delta_c(u) = 0$, allora $\hat{u}$ è strategia ottima;

P4′) se $\operatorname{Max} \Delta_c(u)$ è maggiore di zero, allora sia il valore della ricompensa media relativa alla strategia $\hat{u}$ sia quello relativo alla strategia ottima u^* risultano maggiori di c;
se $\operatorname{Max} \Delta_c(u)$ è minore di zero, allora non esiste alcuna strategia la cui ricompensa media sia superiore a c.

La variante dell'algoritmo consiste nel sostituire la $\Delta_c(u)$ al posto della $\Delta_w(u)$ e nell'introdurre la seguente procedura di tipo sostanzialmente dicotomico.

Ad ogni iterazione, come valore di c da fornire alla SIR, si sceglie il valor medio fra due numeri a e b che inizialmente vengono fissati in modo che la ricompensa media $g(u^*)$ relativa ad una strategia ottima u^* sia sicuramente compresa fra a e b: $a \leqslant g(u^*) \leqslant b$ [9]; successivamente invece vengono fissati tenendo conto dei seguenti tre casi:

I) se $a < g(\hat{u}) < c$ [10] (questo vuol dire (per la P4′) che anche $g(u^*) < c$) è evidente che nella successiva iterazione converrà scegliere $g(\hat{u})$ come nuovo valore di a e c come nuovo valore di b;

II) se $g(\hat{u}) = c$ anche $\hat{u}$ è strategia ottima (per la P3′) e dunque l'algoritmo si arresta;

III) se $c < g(\hat{u}) < b$ (questo vuol dire (per la P4′) che anche $g(u^*) > c$) è evidente che nella successiva iterazione converrà scegliere $g(\hat{u})$ come nuovo valore di a e lasciare inalterato il valore di b.

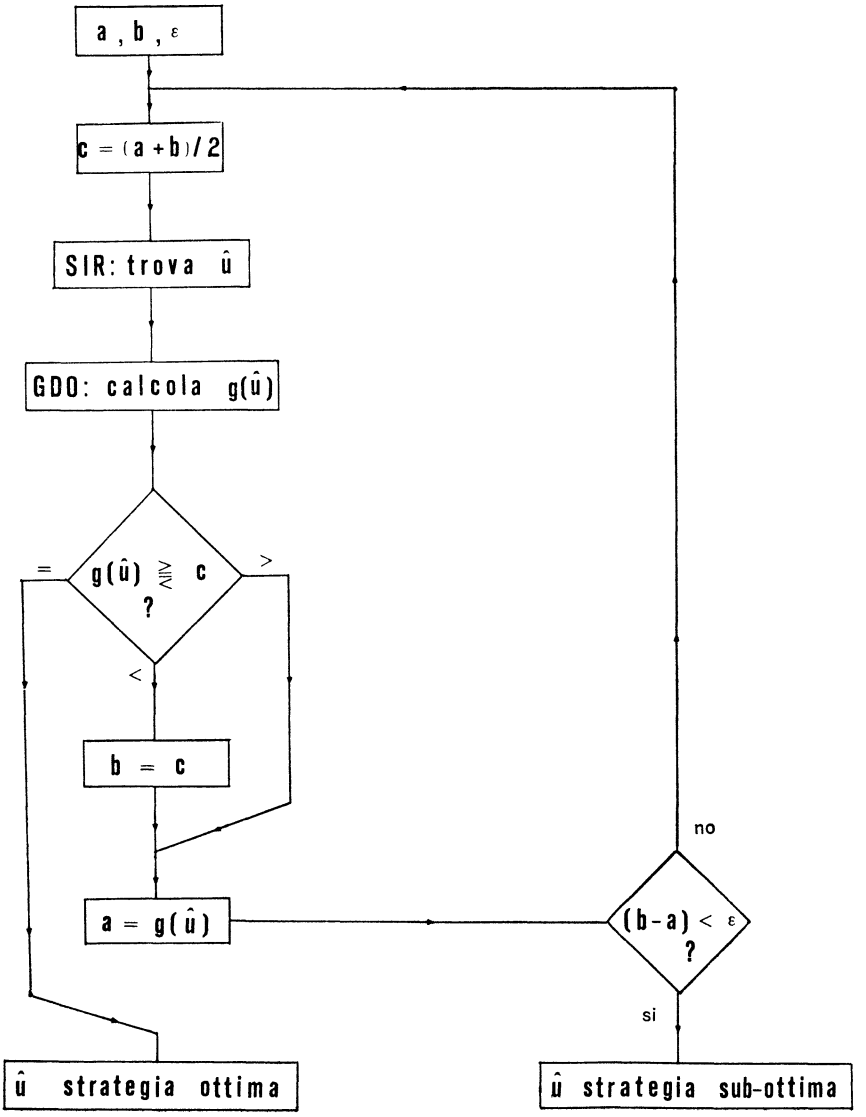
Il numero $(b - a)$ diviene quindi sempre più piccolo: ad ogni iterazione viene più che dimezzato; non appena questo numero risulta inferiore ad ε (se ε è l'approssimazione prescelta) l'algoritmo si arresta: l'ultima strategia fornita dalla SIR è, per quanto visto, la strategia richiesta.

([9]) Per esempio (se le ricompense sono non negative) ponendo

$$a = 0 \qquad \text{e} \qquad b = \operatorname*{Max}_{u \in \mathfrak{U}} \left[\operatorname*{Max}_{0 \leqslant i \leqslant n} q_i(u) \right].$$

([10]) Continuiamo ad indicare con $\hat{u}$ la strategia fornita dalla SIR e con u^* una strategia ottima.

L'algoritmo può quindi essere schematizzato nel modo seguente:



6. Osservazione conclusiva.

Per saggiare il nostro algoritmo abbiamo considerato il « Problema della Manutenzione » (che rientra fra i processi M/M chiusi) prendendo in esame un certo numero di casi specifici (particolarizzando costi e velocità di transizione) e fornendo per ciascuno un « dimensionamento dinamico » del servizio di manutenzione.

I risultati numerici ottenuti (cfr. [1]) confermano la computabilità dell'algoritmo e la sua efficienza per risolvere il problema posto.

Avvertiamo però che, avendo utilizzato una versione dell'algoritmo precedente a quella presentata in questo lavoro, i tempi di calcolo indicati in [1] potrebbero essere considerevolmente diminuiti: nella versione precedente, per calcolare la ricompensa media, si risolveva un sistema lineare utilizzando la « Value-Determination Operation » di Howard anzichè la più semplice GDO qui descritta.

BIBLIOGRAFIA

[1] G. Andreatta, *Dimensionamento ottimale di un servizio di manutenzione*, in corso di pubblicazione su « Ricerca Operativa », Milano.

[2] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, N. J., 1957.

[3] D. Blackwell, *Discrete Dynamic Programming*, Annals of Mathematical Statistics, **33** (1962), pp. 719-726.

[4] R. A. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, Mass. (1960).

[5] M. Volpato, *Nuovi studi e modelli di Ricerca operativa*, Unione tipografico-editrice torinese, Torino (1971).

Manoscritto pervenuto in redazione il 1° aprile 1974.