STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

STATISTISCHE AFDELING

DYNAMISCHE PROGRAMMERING

door

G. de Leve

Lezing te houden op 28 Maart
Voor de Sectie Operations Research
van de V.V.S.

Maart 1961

DYNAMISCHE PROGRAMMERING

Inleiding

Het maken van een scherp onderscheid tussen de dynamische programmering en de overige programmering is geen eenvoudige opgave.
De term Dynamische Programmering is afkomstig van Richard Bellman, die zijn keuze in zijn boek "Dynamic Programming" als volgt rechtvaardigt: [1)]
"The problems we treat are programming problems to use a terminology now popular. The adjective "dynamic", however, indicates that we are interested in processes in which time plays a significant role and in which the order of operations may be crucial".
Nu bestaat er echter een groot aantal zeer ingewikkelde productie problemen, waarin de tijd een zeer belangrijke rol speelt en die met lineaire programmering kunnen worden opgelost. Zijn deze productie problemen nu ook Dynamische Programmerings problemen?
Vervolgens zegt Bellman: "However, an essential feature of our approach will be the reinterpretation of many static processes as dynamic processes in which time can be artificially introduced".
Het hierna volgende is bestemd voor diegenen onder de lezers, die menen dat nadere uitspraken van Bellman nog enige toelichting gewenst is.
Zoals het bij iedere wetenschap aantrekkelijk is om het totaal te bestrijken gebied onder te verdelen in meer gespecialiseerde onderwerpen, zou men gaarne zien dat ook de besliskunde een dergelijke onderverdeling kent. Zover ik weet bestaat er geen aanvaardbare indeling van de besliskunde in onderwerpen en bedient men zich meestal van een nogal zonderlinge opsomming van problemen en methoden. Naast programmerings methoden vindt men b.v. voorraadproblemen, die dikwijls met succes met programmerings methoden kunnen worden opgelost. Een kategorische indeling hetzij naar problemen hetzij naar methoden zou een bijdrage kunnen zijn tot een beter overzicht.

1) R.Bellman, Dynamic Programming, Princeton University Press 1957

Indien men een indeling naar problemen verkiest boven een naar methoden,dan rijst de vraag of men de problemen naar hun physische of, naar hun structurele eigenschappen moet onderscheiden. Tot dus ver spreekt men veelal over vervangings-, voorraad-, productieproblemen etc. en de O.R. man die de volgende dag advies gevraagd zal worden over een nog niet nader geformuleerd voorraadprobleem tast volledig in het duister. Hij weet alleen dat men hem een situatie zal voorschilderen, waarin goederen op één gehoopt staan in stoffige ruimten.

Persoonlijk geef ik de voorkeur aan een indeling naar structurele eigenschappen, die m.i. de enige zinvolle is.Gaat men bovendien uit van de premisse dat de oplossing het belangrijkste aspect van het probleem vormt dan zal, de alternatieve indeling geen groot succes kunnen zijn voor de herkenning van de oplossings methode.

Voor het oplossen van O.R. problemen is het noodzakelijk dat men door het probleem kan heen zien om te ontdekken hoe de "machinerie" werkt. Met behulp van een mathematisch model kunt U de hoofdlijnen van de structuur van het probleem vast leggen.

Uit een en ander volgt, dat structureel gezien sommige transportproblemen en sommige personeel klassificatie problemen identiek zijn, terwijl dergelijke overeenkomsten ook bestaan tussen voorraadproblemen enerzijds en wachttijd- en vervangingsproblemen anderzijds.

Zonder een volledige structurele indeling te geven van de besliskunde meen ik dat de groep "Multi stage decision processes" een eenheid vormt.

Problemen die tot deze groep behoren kunnen veelal worden opgelost met behulp van één of meer van de volgende methoden:

a) lineaire of niet-lineaire programmering
b) dynamische programmering
c) methoden waarbij men gebruik maakt van de theorie der Markov processen
d) methoden waarbij men gebruik maakt van de theorie van het Servo-Mechanisme

etc.

Aan welk van de methoden men de voorkeur zal geven hangt af van het wiskundige model en de technische hulpmiddelen,b v. rekenmachines, die men tot zijn beschikking heeft.

Multi stage decision processes

Ter vereenvoudiging van de discussie zullen wij eerst een aantal begrippen omschrijven, die wij in het hierna volgende meer dan eens zullen gebruiken.
Het eerste begrip, dat wij zullen invoeren heet systeem. Het systeem is veelal het object van onze beschouwingen. Zo kan bij een wachttijdprobleem het systeem gevormd worden door de rij wachtenden voor een loket. Bij een voorraadprobleem zal met het systeem meestal de voorraad worden bedoeld.
Aangenomen wordt dat het systeem zich in verschillende toestanden kan bevinden en dat deze toestanden beschreven kunnen worden door een aantal kwantitatieve grootheden. Een toestand wordt dus gegeven door een rij van getallen, de waarde van de kwantitatieve grootheden voor deze toestand.
Een dergelijke rij van getallen noemt men een toestandsvector. Indien wij een voorraad moeten beheren van drie verschillende artikelen dan is het systeem dus een voorraad. De toestand van het systeem, zo zullen wij aannnemen, kan worden vastgelegd door drie getallen, die ieder de omvang van de voorraad van één artikel aangeven. Stel dat de toestanden van een systeem beschreven kunnen worden met behulp van m kwantitatieve grootheden, dan kan iedere toestand geidentificeerd worden met een punt in een door m orthogonale coördinaatassen opgespannen ruimte of met de toestandsvector, die de verbinding van dat punt met de oorsprong van het assenstelsel is. Op de assen van het assenstelsel worden de eerdergenoemde kwantitatieve grootheden uitgezet. Deze ruimte zullen wij de toestandsruimte noemen. In ieder probleem, dat tot deze klasse behoort, moet op een eindig aantal discrete tijdstippen een beslissing worden genomen. Indien in een probleem op n discrete tijdstippen een beslissing genomen dient te worden, dan spreekt men van een n-staps beslisprobleem. Zodra de eerste beslissing genomen is gaat het n-staps probleem over in een (n-1)-staps beslisprobleem. We zullen aannemen, dat ook iedere mogelijke beslissing vastgelegd kan worden met behulp van een aantal kwantitatieve grootheden.
In het hierboven genoemde voorraadprobleem kan iedere beslissing worden vastgelegd door drie getallen. Deze getallen geven de order grootten aan van de drie artikelen.

Voor het geval men de beslissingen kan aanduiden met behulp van r kwantitatieve grootheden, correspondeert met iedere beslissing een punt in een r dimensionale ruimte. De verbindingslijn van dat punt met de oorsprong zullen wij de beslissingsvector noemen, terwijl de ruimte zelf met beslissingsruimte wordt aangeduid.
Bij een zorgvuldige beschouwing van de problemen uit deze klasse valt het ons op, dat de toestand van het systeem op het volgende tijdstip van beslissen sterk wordt beinvloed door de toestand op het huidige tijdstip van beslissen en door de beslissing zelf en echter niet door de toestanden en beslissingen op voorgaande tijdstippen. Dit laatste zal zonder meer duidelijk zijn, als men weet dat in het begrip toestand alle relevante informatie over het systeem nu en in de toekomst is verwerkt. Het eerste vereist echter enige toelichting. Bij een definiet proces kan men nagaan wat de gevolgen zijn van een beslissing in een bepaalde toestand. Bij een proces met stochastische eigenschappen kan men niet met zekerheid voorspellen wat de gevolgen zijn van een beslissing in één of andere toestand. Wel zal de kansverdeling gedefiniëerd op de verzameling van alle mogelijke resultaten slechts afhangen van de beslissing en de toestand waarin de beslissing genomen is.
Verder kunnen wij constateren, dat ten gevolge van een beslissing steeds sprake is van een opbrengst.[1] De omvang van deze opbrengst hangt af van de genomen beslissing en de toestand waarin deze beslissing is genomen.
Een reeks van beslissingen zullen wij optimaal noemen als de totale opbrengst maximaal is.[2]
Verder blijkt dat wij voor de bepaling van de optimale beslissing op één tijdstip niet kunnen volstaan door alleen maar de opbrengst te beschouwen, welke een direct gevolg is van de beslissing. Immers de beslissing bepaalt mede de toestand op het volgende beslissingstijdstip en dientengevolge de toekomstige opbrengsten.

1) Er zijn problemen, waarbij de opbrengst stochastisch is. In die gevallen beschouwt men de verwachting van de opbrengst.

2) Er zijn problemen, waarbij niet de totale opbrengst maximaal gekozen moet worden maar een gewogen som van opbrengsten. Indien de "opbrengst" een verlies is dan wordt uiteraard geminimaliseerd.

Om het effect van een beslissing op toekomstige opbrengsten te kunnen nagaan, dient men, omdat de opbrengsten hierdoor mede worden bepaald, de toekomstige beslissingen te kennen.

N-staps dynamisch programmerings probleem

Aan deze moeilijkheid kan op de volgende wijze het hoofd worden geboden. Men begint nl. bij een n-staps d.p. probleem niet met de bepaling van de eerste optimale beslissing maar met de laatste. Hiervoor dient men de toestand te kennen op het bijbehorende tijdstip. Aangezien men die toestand niet kent lost men het probleem op voor alle mogelijke toestanden. De maximale opbrengst is derhalve een functie van de toestand op het laatste beslissingstijdstip. Deze toestanden zelf of de kansverdeling behorende bij deze toestanden worden volledig bepaald door de toestand van het systeem en de beslissing op het voorgaande moment van beslissen. De maximale opbrengst kan dus beschouwd worden als een functie alleen van die grootheden. Indien men een nieuwe opbrengstfunctie invoert, die gelijk is aan de oude vermeerderd met de maximum opbrengst van de volgende beslissing, dan kan deze nieuwe opbrengstfunctie gemaximaliseerd worden, waardoor de optimale beslissing voor het op één na laatste moment van beslissen wordt verkregen.

Ook nu geldt dat de toestand van het systeem op het beslissingstijdstip niet bekend is zodat het maximum een functie is van deze toestand.

Resumerend kunnen wij vaststellen, dat wij eerst 1-staps problemen oplossen voor alle mogelijke uitgangstoestanden van het systeem en daarna met behulp van deze oplossingen 2-staps problemen oplossen voor alle mogelijke begintoestanden.

Indien men op deze wijze voortgaat, dan wordt de oplossing van het oorspronkelijk gestelde vraagstuk verkregen.

Bij het oplossen van een d.p. probleem wordt het oorspronkelijke probleem dus vervangen door een reeks van in eenvoud afnemende problemen, welke worden opgelost voor alle mogelijke begintoestanden.

Indien wij de toestand van het systeem en de beslissing op het $m^{\underline{de}}$ beslissingstijdstip resp. aangeven met x_m en d_m en als wij verder schrijven voor de bijbehorende opbrengst $h(x_m, d_m)$ en de toekomstige[3] opbrengsten $H_{n-m}(x_m, d_m)$ bij optimale keuze van d_{m+1}, d_{m+2} enz. dan geldt voor een n-staps d.p. probleem:

3) evt. gewogen

$$H_{n-m}(x_m, d_m) = \max_{d_{m+1}} \left[h(x_{m+1}, d_{m+1}) + H_{n-m-1}(x_{m+1}, d_{m+1})\right] \quad (m \leq n-1) \quad (1)$$

waarbij

$$H_o(x_n, d_n) = 0 \quad (2)$$

Uit deze beschouwing volgt tevens, dat bij een n-staps d.p. probleem de eerste optimale beslissing d_1 wordt bepaald door de toestand x_1. Men zou ook kunnen zeggen dat door de optimale keuze aan een toestandsvector in de toestandsruimte een beslissingsvector in de beslissingsruimte wordt toegevoegd. Deze toevoeging kan men ook als volgt uitdrukken:

$$d_1 = S_n(x_1). \quad (3)$$

Bellman heeft door gebruik te maken van het abstracte begrip Functieruimte de zgn. strategieruimte ingevoerd. Ieder punt uit deze ruimte is een functie met behulp waarvan aan iedere toestandsvector een beslissingsvector kan worden toegevoegd. Ook de functie $S_n(x)$ in (3) is een punt uit deze ruimte en wordt een strategie-functie of ook wel kortweg strategie genoemd.

∞-staps dynamisch programmerings probleem

Wij zullen ons nu bezig houden met situaties, waarin het aantal te nemen beslissingen onbegrensd is. Het is zonder meer duidelijk, dat de oplossing van deze problemen niet verkregen wordt door eerst de beslissing van de "laatste" stap te bepalen.

Stel dat de maximale opbrengst bij een n-staps d.p. probleem gegeven wordt door de functie $f_n(x)$, waarbij x de toestandsvector aanduidt op het eerste tijdstip van beslissen.

Uit de definitie van $H_i(x,d)$ volgt:

$$f_n(x) = \max_d \left\{ h(x,d) + H_{n-1}(x,d)\right\} \quad (4)$$

Men kan nu de volgende gevallen onderscheiden: [1]

a) Uit de probleemstelling volgt, dat de toestand van het systeem op het volgende beslissingstijdstip volledig bepaald wordt door de toestand op het huidige moment van beslissen en de beslissing zelf.

b) Uit de probleemstelling volgt, dat de kansverdeling van de toestandsvector behorende bij het volgende beslissingstijdstip volledig bepaald wordt door de toestand op het huidige moment van beslissen en de beslissing zelf.

1) De situatie, welke onder a) wordt beschreven, kan beschouwd worden als een speciaal geval van die vermeld onder b).

Onafhankelijk hiervan kunnen zich nog de volgende situaties voordoen [1)]

c) Uit de probleemstelling volgt, dat bij de i^{de} beslissing met de volledige opbrengst te verkrijgen uit toekomstige beslissingen rekening wordt gehouden.

d) Uit de probleemstelling volgt, dat op het i^{de} beslistijdstip slechts rekening wordt gehouden met een fractie van de opbrengsten behorende bij toekomstige beslissingen en wel zo dat de bijdrage van de j^{de} beslissing ($j \geq i$) tot de totale opbrengst gelijk is aan:

$$\alpha^{j-i}\, h(x_j;\, d_j) \tag{5}$$

Schrijft men nu voor de toestandsvector en beslissingsvector behorend bij het i^{de} beslissingstijdstip, x_i resp. d_i, dan vinden wij voor de opbrengstfunctie $f_n(x)$:

I $$f_n(x_1) = \max_{d_1} \left\{ h(x_1, d_1) + f_{n-1}\left[x_2(x_1, d_1)\right]\right\} \tag{6}$$

als zowel a) als c) van toepassing is.

II $$f_n(x_1) = \max_{d_1} \left\{ h(x_1, d_1) + \alpha f_{n-1}\left[x_2(x_1, d_1)\right]\right\} \tag{7}$$

als zowel a) als d) van toepassing is.

III $$f_n(x_1) = \max_{d_1} \left[h(x_1, d_1) + E\left\{ f_{n-1}(\underline{x}_2)/\, x_1; d_1 \right\}\right] \tag{8}$$

als zowel b) als c) van toepassing is.

IV $$f_n(x_1) = \max_{d_1} \left[h(x_1, d_1) + \alpha E\left\{ f_{n-1}(\underline{x}_2)/\, x_1; d_1 \right\}\right] \tag{9}$$

als zowel b) als d) van toepassing is.

Beschouwen wij nu de rij van functies $f_n(x)$ dan zullen, als deze rij voor iedere x convergeert tot een functie f(x), de relaties (6) t/m (9) overgaan in: [2)]

I $$f(x_1) = \max_{d_1} \left\{ h(x_1, d_1) + f\left[x_2(x_1, d_1)\right]\right\} \tag{10}$$

II $$f(x_1) = \max_{d_1} \left\{ h(x_1, d_1) + \alpha f\left[x_2(x_1, d_1)\right]\right\} \tag{11}$$

III $$f(x_1) = \max_{d_1} \left[h(x_1, d_1) + E\left\{ f(\underline{x}_2)/\, x_1; d_1 \right\}\right] \tag{12}$$

IV $$f(x_1) = \max_{d_1} \left[h(x_1, d_1) + \alpha E\left\{ f(\underline{x}_2)/\, x_1; d_1 \right\}\right] \tag{13}$$

Indien de opbrengstfuncties $f_n(x)$ behorende bij de overeenkomende n-staps d.p. problemen convergeren tot de functie f(x), dan wordt

1) De situatie, welke onder c) wordt beschreven, kan beschouwd worden als een speciaal geval van die vermeld onder d).

2) De relaties (10) t/m (12) zijn byzondere gevallen van (13)

de oplossing van het ∞-staps d.p. probleem verkregen uit de oplossing van de corresponderende functionaalvergelijking [(10) t/m (13)].
Het onderzoek naar de ondubbelzinnigheid en de existentie van de oplossingen van deze functionaalvergelijkingen vormt een wezenlijk deel van de dynamische programmering. Aan dit belangrijke onderdeel van de theorie der dynamische programmering zullen wij echter geen aandacht besteden.[1)]
Indien echter bij een ∞-staps d.p. probleem de totale opbrengst gezien vanuit de toestand op het eerste beslissingstijdstip voor meer dan één reeks van beslissingen onbegrensd groot wordt, dan kan men de gewenste oplossing niet met behulp van bovenstaande functionaalvergelijkingen verkrijgen. Men zal in zo'n situatie trachten een nieuw kriterium op te stellen om alsnog objectief de resultaten van deze verschillende reeksen van beslissingen te kunnen onderscheiden.
Soms is het zinvol te beschouwen de rij van functies [2)]

$$g_n(x) = \frac{f_n(x)}{n} \tag{14}$$

De functies $g_n(x)$ stellen de gemiddelde winst per beslissing voor bij een n-stap d.p. probleem. Bij menig probleem convergeert deze rij van functies tot een functie $g(x)$.

<u>1ste iteratie methode</u>. Indien de rij van opbrengstfuncties $f_n(x)$ van de overeenkomende n-staps d.p. problemen convergeert voor iedere x tot de opbrengstfunctie $f(x)$ van het ∞-staps d.p. probleem, dan kan deze functie uiteraard iteratief worden bepaald. Heeft men de functie $f(x)$ gevonden dan kan men met behulp van één der relaties (10) t/m (13) voor iedere toestandsvector x de optimale beslissing d bepalen.
Indien de rij van opbrengstfuncties $f_n(x)$ van de overeenkomende n-staps d.p. problemen <u>niet</u> convergeert, maar wel de rij van strategiefuncties $s_n(x)$ [zie (3)] dan kan men ondanks de divergentie van de rij $f_n(x)$ toch de optimale strategie

$$S(x) = \lim_{n \to \infty} S_n(x) \tag{15}$$

bepalen. Veelal wenst men slechts de optimale strategie te kennen en is dus hiermede het probleem opgelost. In dergelijke situaties komt het ook veel voor dat de limiet

1) Wij verwijzen hiervoor naar R.Bellman: Dynamic Programming blz. 116 e.v.

2) Wij gebruiken in dit hoofdstuk vaak de termen soms, veelal, etc. zonder precies aan te geven onder welke voorwaarden de uitspraak waar is. Ook hiervoor geldt 1).

$$\lim_{n\to\infty} \frac{f_n(x)}{n} = \lim_{n\to\infty} g_n(x) = g(x) \tag{16}$$

bestaat.

2de iteratie methode. Indien voor iedere strategiefunctie $S(x)$ eenvoudig de bijbehorende opbrengst $f(x)$ kan worden bepaald, dan kan men dikwijls ook op een andere iteratieve wijze de functionaalvergelijkingen (10) t/m (13) oplossen. Men begint dan met een willekeurige strategiefunctie $S^{(0)}(x)$, waarvoor de bijbehorende opbrengst $f^{(0)}(x)$ wordt bepaald. Voor de strategiefunctie $d_1 = S^{(1)}(x)$ geldt nu:

$$\text{I} \quad h(x_1,d_1) + f^{(0)}\left[x_2(x_1,d_1)\right] \tag{17}$$

$$\text{II} \quad h(x_1,d_1) + \alpha f^{(0)}\left[x_2(x_1,d_1)\right] \tag{18}$$

$$\text{III} \quad h(x_1,d_1) + E\left\{f^{(0)}(\underline{x}_2)/\ x_1,d_1\right\} \tag{19}$$

$$\text{IV} \quad h(x_1,d_1) + \alpha E\left\{f^{(0)}(\underline{x}_2)/\ x_1,d_1\right\} \tag{20}$$

is maximaal. De bijbehorende opbrengstfunctie $f^{(1)}(x)$ kan met behulp van $S^{(1)}(x)$ worden bepaald. Indien men op deze wijze voortgaat maximaliseerd de strategiefunctie $d_1 = S^{(i)}(x)$ de uitdrukkingen

$$\text{I} \quad h(x_1,d_1) + f^{(i-1)}\left[x_2(x_1,d_1)\right] \tag{21}$$

$$\text{II} \quad h(x_1,d_1) + \alpha f^{i-1}\left[x_2(x_1,d_1)\right] \tag{22}$$

$$\text{III} \quad h(x_1,d_1) + E\left\{f^{(i-1)}(\underline{x}_2)/\ x_1,d_1\right\} \tag{23}$$

$$\text{IV} \quad h(x_1,d_1) + \alpha E\left\{f^{(i-1)}(\underline{x}_2)/\ x_1,d_1\right\} \tag{24}$$

Uit de constructie verkrijgen wij de rijen van functies $S^{(i)}(x)$ en $f^{(i)}(x)$ $(i=1,\ldots)$. Indien de rij $f^{(i)}(x)$ convergeert en men weet dat de oplossing van de functionaalvergelijkingen (10) t/m (13) bestaat en ondubbelzinnig is, dan kan men ook op deze wijze de optimale strategie bepalen. Deze oplossingsmethode wordt ook wel eens aangegeven met de naam Approximatie in de strategieruimte.

Markovian decision processes.

Wij beschouwen wederom een ∞-staps beslisprobleem, waarbij men op van te voren vastgestelde tijdstippen een beslissing kan nemen. Indien alle relevante informatie over het systeem nu en in de toekomst verwerkt is in het begrip toestand dan zal men elke keer als het systeem op een beslissingstijdstip in een bepaalde toestand komt de zelfde beslissing nemen. Hieruit volgt dat men de optimale strategie kan vastleggen door aan ieder punt uit de toestandsruimte een punt uit de beslissingsruimte toe te voegen. Deze beslissing wordt slechts dan uitgevoerd als het systeem deze toestand aanneemt op een beslissingstijdstip. Nu bestaat er een uitgebreide klasse van problemen, waarvan de toestandsveranderingen van het systeem beschreven kunnen worden door een Markov proces. Dit Markov proces hangt af van de keuze van de strategie. Voor verschillende strategieen krijgt men verschillende Markov processen. Ter vereenvoudiging van de discussie zullen wij aannemen dat het systeem zich slechts in eindig veel toestanden i kan bevinden $(i=1,\ldots\ldots n)$.
Stel dat de kans op een overgang van toestand i nu naar een toestand j op het volgende beslissingstijdstip gegeven wordt door $p_{ij}(x)$, als x de toegepaste strategie voorstelt. Indien $p_{ij}^{(k)}(x)$ de kans voorstelt op een overgang van i nu naar een toestand j op k beslissingstijdstippen verder dan geldt:[1]

$$p_{ij}^{(k)} = \sum_{l=1}^{n} p_{il}^{(k-1)} \, p_{lj} \tag{25}$$

Nu kan men bewijzen dat de uitdrukking

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} p_{ij}^{(k)} = q_{ij}(x) \tag{26}$$

bestaat. Men noemt de grootheden q_{ij} de invariante kansen van het Markov proces.
Stel dat $h(i,x)$ de opbrengst is, wanneer men in de toestand i een beslissing neemt volgens X. De verwachting van de totale opbrengst in m stappen wordt dan gegeven door:

$$\sum_{k=1}^{m} \sum_{j=1}^{n} p_{ij}^{(k)} h(j;\, x) \tag{27}$$

1) Wij laten voortaan de vermelding van de strategie achterwege.

Indien men een strategie zoekt waarvoor de opbrengst voor de eerste m stappen maximaal is, dan moet voor deze strategie gelden dat (27) maximaal is. Het maximaliseren van de uitdrukking (27) is gelijkwaardig met het maximaliseren van

$$\frac{1}{m} \sum_{k=1}^{m} \sum_{j=1}^{n} p_{ij}^{(k)} h(j;\chi) = \sum_{j=1}^{n} \left[\frac{1}{m} \sum_{k=1}^{m} p_{ij}^{(k)} \right] h(j;\chi) \qquad (28)$$

Indien $m \to \infty$ volgt uit (28) en (26) dat men als criterium moet kiezen voor de optimale strategie:

$$R(\chi) = \sum_{j=1}^{n} q_{ij}(\chi) h(j;\chi) \qquad (29)$$

Met behulp van (29) kan men dus twee alternatieve strategieen vergelijken.
Soortgelijke uitdrukkingen verkrijgt men ook als het systeem zich in meer dan eindig veel toestanden kan bevinden.

Indien men vrij in de keuze van het beslissingstijdstip is, dan wordt de procedure iets ingewikkelder. Maar ook dan kan men veelal met behulp van Markov processen de oplossingen verkrijgen van ∞ -staps beslisproblemen.