

# HIERARCHIKUS MARKOV FOLYAMATOK ALKALMAZÁSA A SERTÉSTARTÁS DÖNTÉSI FOLYAMATAIBAN

## APPLICATION OF THE HIERARCHIC MARKOVIAN DECISION PROCESSES IN THE DECISION MAKING PROCESSES OF PIG KEEPING

Kovács Sándor<sup>1</sup>, Balogh Péter<sup>2</sup>

**Absztrakt:** Jelen tanulmány keretén belül tárgyaljuk a Markov láncokon alapuló Markov döntési folyamatokat, valamint ezek továbbfejlesztett változatát, a Hierarchikus Markov Folyamatokat. Részletesen leírjuk ezen folyamatok optimalizációs lehetőségeit, valamint ismertetünk egy dán kutatók által kifejlesztett, e folyamatokon alapuló ingyenes szoftvert, melyet kifejezetten állattenyésztési döntések támogatására alkottak. A számos lehetőség közül a koca nevelési és vemhesítési modellt mutatjuk be. Megadjuk a program futtatásához szükséges alapparamétereket, beállításokat, valamint egy szimulált telep összes kocájára érték iterációval kiszámítjuk az elérhető jövedelmet és a kocánkénti döntések sorozatát. Bemutatjuk annak a lehetőségét, hogy az eredmények alapján hogyan dönthetünk például olyan kérdésekről, hogy termelésben tartsuk-e az állatot, vagy ne, illetve hányszor érdemes termékenyíteni a kocát az élete folyamán. Példát adunk a kapcsolódó relatív hasznosságok alakulására, ha adott döntést hozunk egy másikkal szemben.

**Kulcsszavak:** Markov folyamatok, koca modell, vemhesítés, gazdaságosság

### 1. Bevezetés

A mikroszámítógépek elterjedése megnövelte az informatika jelentőségét az állattenyésztésben is, különösképpen az információs és döntéstámogató rendszerek kifejlesztésében. A döntéstámogató rendszerek állattenyésztésben történő alkalmazása még a fejlett országokban sem éri el a szakemberek által megkövetelt szintet, ezért ez egy kedvelt és visszatérő téma az informatikai konferenciákon. A tervezés bonyolultsága szükségessé teszi matematikai módszerek kifejlesztését, alkalmazását a vizsgált problémák megoldásának megkönnyítésére. Erre a célra a mezőgazdaságban számos matematikai módszert vettek már igénybe, mint például a hálótervezés, a lineáris programozás, függvény analízis és a matematikai statisztika. Ezek az analitikus módszerek az optimális megoldásra törekszenek, ezért azonban árat kell fizetnünk. A lineáris programozás esetében nagy kötöttséget jelen az, hogy lineáris összefüggéseket keresünk, és ezzel leszűkítjük az alkalmazás területeit. A járhatóbb út a valóság modellezése úgynevezett szimulációs kísérlettel, mely nagyobb rugalmasságot biztosít és kiszélesíti a felhasználás körét. A szimulációs módszer elsősorban nem optimalizációs eljárás, de az általunk bemutatott módszer, amely a dinamikus programozásból nőtte ki magát beépített optimalizációs modullal is rendelkezik. Jelen tanulmány keretén belül bemutatjuk a Hierarchikus Markov Folyamatok részletes módszertanát, azok optimalizációs lehetőségeit, valamint ismertetünk egy dán kutatók által kifejlesztett, ingyenes szoftvert, melyet kifejezetten állattenyésztési döntések támogatására alkottak. A számos lehetőség közül egy alapvető problémát tárgyalunk, a kocák vemhesítésének problémáját. A program bemutatása során a módszertanban bemutatott fogalmakra támaszkodunk.

### 2. Történeti áttekintés

A dinamikus programozás módszertanát először Bellman publikálta. (Bellman,1957). A könyvben a szekvenciális döntési problémák megoldására szolgáló, új numerikus módszert adott közre, melynek fő elemei a „Bellman-féle optimalitási követelmények” és a funkcionális egyenletek. Az első művét követően Bellman még számos más könyvet adott ki a témában (Bellman, 1961), (Bellman and

<sup>1</sup> Debreceni Egyetem, kovacs@agr.unideb.hu

<sup>2</sup> Debreceni Egyetem, balogh@agr.unideb.hu

Dreyfus, 1962), (Bellman és Kalaba, 1965). A dinamikus programozásról sokan azt tartották, hogy egy triviális számolási eszköz, mások túlságosan sokra értékelték. Később azonban igazolódott, hogy sem triviális, sem pedig a valós élet problémáinak széles köre nem oldható meg vele. Fő eleme a dinamikus programozásnak a szekvenciális megközelítési mód, ami azt jelenti, hogy a sorozatos döntési problémákra illeszkedik leginkább. A legnyilvánvalóbb példa sorozatos döntési problémákra az állattenyésztési ágazatban az állat selejtezésének, illetve új kocasüldő beállításának folyamata, valamint az inszeminálás, vemhesítés és orvosi ellátás. A módszer azért is releváns az állattenyésztési problémákra, mert az állat jellemzőit (pl.: alacsony vagy magas hozam, megtermékenyülési képesség, alomszám, stb.) valószínűségekkel becsülni tudjuk, és döntéseink a jellemzők aktuális megfigyeléseire vonatkoznak. A dinamikus programozás Markov folyamatokkal történő összekapcsolását Howard dolgozta ki (Howard, 1960). A két módszer kombinálásával jött létre a Markov döntési folyamat (MDF) fogalma, melyet szintén Howard vezetett be. Ő ismertette a politika iterációt, és az érték iterációs technikát is, amellyel az MDF optimalizálható.

A politika iterációt két optimalitási kritérium miatt fejlesztették ki, nevezetesen a teljes időhorizontra vonatkozó, elvárt diszkontált jövedelem és az elvárt átlagos időszakos jövedelem maximalizálására. Később Jewell mutatta be a politika iterációt az átlagos jövedelem maximalizálására a teljes időhorizonton, és semi-Markov döntési folyamatnak nevezte (Jewell, 1963). Ez egy olyan MDF, amelyben az időszakok hosszai véletlen változók. Howard az értékiterációra is kidolgozta a semi-Markov döntési folyamatokat (Howard, 1971). A teljesség kedvéért megjegyzendő, hogy a lineáris programozás módszerét már az említett két optimalizációs technika előtt alkalmasnak tartották az MDF-ek optimalizálására (Hadley, 1964), de semelyik állattenyésztési modellben nem alkalmazták a kutatók ezt a technikát, mivel a politika iteráció hatékonyabbnak bizonyult a lineáris programozásnál. Howard 1960-ban publikált könyve után számos olyan kutatás látott napvilágot, amely az optimalizációs technikák és az optimalitási kritériumok kapcsolatát vizsgálta (Wal and Wessels, 1985). Az MDF-es alkalmazások domináns területe az állatok utánpótlásának, selejtezésének a problémája összekapcsolva az orvosi kezelésekkal és az inszeminálással. Az eddigi módszertani fejlesztések nagy részét Kristensen és Jorgensen (1995) publikálta. Az alkalmazások 3 fő nehézsége a változékonyság, reprodukív ciklusok, korlátok (például a malacnevelő képesség határai, vagy a véges elhelyezési kapacitás). Az első két nehézséget leküzdí az MDF, az utóbbin pedig nem lehet változtatni.

### 3. Anyag és módszertan

#### 3.1. Markov lánc

A Markov döntési folyamatok alapja a Markov lánc fogalma, ezért megértésükhöz feltétlenül szükség van néhány alapvető fogalom tisztázására a Markov láncok elméletéből. A Markov lánc a sztochasztikus folyamatok speciális típusa. Egy diszkrét idejű sztochasztikus folyamatot akkor nevezünk Markov láncnak, ha a  $P(X_{t+1} = i_{t+1} | X_t = i_t)$  feltételes valószínűség fennáll.

Mindez azt jelenti, hogy a  $t+1$ -edik időszak állapotának valószínűségeloszlása a  $t$ -edik időszak állapotától függ, de nem függ azon korábbi időszakoktól a láncban, amelyeken már áthaladt a folyamat. Ebben az értelemben általánosan feltehetjük, hogy minden  $i$  és  $j$  állapotra minden  $t$  időszakban fennállnak a  $P(X_{t+1} = j | X_t = i) = p_{ij}$  és  $P(X_{t+1} = j | X_t = i)$  valószínűségek és ezek csak  $t$ -től függenek. A  $p_{ij}$  jelöli annak a valószínűségét, hogy az adott rendszer a  $t$  időszaki  $i$ -edik állapot után a  $t+1$ -edik időszak  $j$ -edik állapotába kerül. Innen ered  $p$ -re az átmenet valószínűség elnevezés is. Továbbá azok a valószínűségek, amelyek az első időszak  $i$ -edik állapotának bekövetkezésére vonatkoznak, azok a kezdő valószínűségeloszlások, jelük  $q_i$ . Így  $P(X_1 = i) = q_i$ .

Végül minden  $t$  időszaki  $i$  állapotra teljesül a következő egyenlet:

$$\sum_j P(X_{t+1} = j | X_t = i) = 1$$

Ebből az is következik, hogy az átmenet valószínűségek alkotta mátrix elemei nemnegatívak, és minden sorösszeg 1-et ad. A  $p_{ij}$  valószínűségeket gyakran csak egy lépéses valószínűségeknek

nevezik, mivel a  $t$  időszaktól  $t+1$ -edik időszakra térünk át. Szokásos erre a  $p_{ij}(1)$  jelölést használni. Minden Markov láncsal megoldható problémánál az a kérdés merül fel, hogy mi a valószínűsége annak, hogy  $j$  állapotban lesz a folyamat  $n$  lépés múlva, ha jelenleg az  $i$ -edik állapot jellemzi. Markov lánc esetén a stacionaritását használjuk ki, ami azt jelenti, hogy a valószínűségek függetlenek  $t$  időszaktól. Ebből az adódik, hogy:

$$P(X_{t+1} = j | X_t = i) = P(X_n = j | X_0 = i) = p_{ij}(n)$$

ahol a  $p_{ij}(n)$  az úgynevezett  $n$  lépéses valószínűség. Természetesen  $p_{ij}(1) = p_{ij}$  és  $p_{ij}(2) = \sum_k p_{ik} \cdot p_{kj}$ .  $p_{ij}(2)$  az  $ij$ -edik eleme a  $P^2$  mátrixnak, ahol  $P = p_{ij}(1)$  az átmenet valószínűségek mátrixa. Általánosan pedig  $p_{ij}(n)$  az  $ij$ -edik eleme a  $P^n$  mátrixnak. Általában a legtöbb Markov lánc esetén többnyire létezik a  $\pi = (\pi_1, \pi_2, \dots, \pi_s)$  vektor (Barbour, 2004):

$$\lim_{n \rightarrow \infty} P^n = \begin{bmatrix} \pi_1 & \pi_2 & \dots & \pi_s \\ \pi_1 & \pi_2 & \dots & \pi_s \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \pi_1 & \pi_2 & \dots & \pi_s \end{bmatrix}$$

vagy másképpen:  $\lim_{n \rightarrow \infty} p_{ij}(n) = \pi_j$ .

Ez azt jelenti, hogy a Markov lánc állandósul a  $j$ -edik állapotban  $\pi_j$  valószínűséggel,  $s$  ez független a kiinduló állapottól. A  $\pi$  vektor a Markov folyamat egyensúlyi eloszlása. Két módja is van az egyensúlyi eloszlás megtalálásának. Az egyik mód az, hogy az átmenet valószínűségi mátrixot önmagával tetszőlegesen sokszor összeszorozzuk. A másik út pedig az egyenletrendszerekkel történő meghatározás. A  $P$  átmenet valószínűség mátrix elemei képezik az egyenletrendszerek ismeretlenjeinek együtthatóit, az ismeretlenek a vektor elemei, az egyenlet jobboldalán szintén a  $\pi$  vektor elemei állnak.

### 3.2. Markov döntési folyamatok

Legyen adott egy rendszer, amelyet véges vagy végtelen időhorizonton figyelünk meg. Az időhorizont periódusokra vagy időszakokra osztható. Minden egyes időszakban a rendszert adott állapotában figyeljük meg és szükség szerint beavatkozunk. A döntés vagy determinisztikusan vagy sztochasztikusan befolyásolja az állapotot, amelyet a következő időszakban figyelünk meg. A rendszer állapota és az időszak függvényében egy ennek megfelelő jövedelmet érünk el. A teljes elvart jövedelmet az adott időszaktól kezdve egészen a tervezési időhorizont végéig egy úgynevezett értékfüggvény adja meg. A kapcsolatot az adott időszak értékfüggvénye és a következő időszak számított értékfüggvénye között a funkcionális egyenletek adják meg. Az optimális döntés, amely az időszaktól és az állapottól függ, úgy határozható meg, hogy lépésről lépésre visszafelé haladva maximalizálni kell a funkcionális függvény jobboldalát.

Legyen adott egy diszkrét Markov döntési folyamat egy véges állapottérrel,  $U = \{\text{állapot } i = 1, 2, \dots, u\}$  és egy véges  $D$  döntési halmazzal. Az  $s$  politika pedig egy döntési fa, speciális elrendezés, amely minden egyes  $i$  állapothoz egy döntést rendel, tehát  $s(i) = d \in D$ . Legyen  $p_{ij}^d$  az  $i$  állapotból  $j$  állapotba kerülés úgynevezett átmenet valószínűsége, amikor  $d \in D$  döntést hoztuk. A  $d$  döntés által nyerhető jövedelem  $i$  állapotban  $r_i^d$ . A két átmenet közötti időintervallumot időszakknak nevezzük. Néhány modell az  $m_i^d$  fizikai mennyiséget (egy anyakocára jutó alomnagyság, vagy élettartam) is tartalmazza  $i$  állapotban  $d$  döntéstől függően (Kristensen, 1991). A  $p_{ij}^d$ ,  $r_i^d$ ,  $m_i^d$  jelölések mellett  $p_{ij}^s$ ,  $r_i^s$ ,  $m_i^s$  jelölés is használatos és elfogadott, ha  $s(i) = d$ . Az optimális politika maximalizál egy előre definiált célfüggvényt. Az optimalizációs technika függ a célfüggvény alakjától, amelyet másképpen

optimalizációs kritériumnak is neveznek. A kritériumok megválasztása attól is függ, hogy az időhorizont véges-e, vagy végtelen.

### 3.3. Optimalizálási kritériumok

Tegyük fel, hogy adott T időszak, tehát véges időhorizonton mozgunk, és a teljes időhorizontra vonatkozó, várható jövedelmet maximalizálni akarjuk. Ekkor a következő célfüggvényt alkalmazzuk:

$$Hozam_1(s_1, \dots, s_T) = E\left(\sum_{i=1}^T r_{I_i}^{s_i}\right),$$

Ahol az „E” a várhatóérték,  $s_i$  az i-edik időszaki politika,  $I_i$  pedig a nem ismert állapota az i-edik időszaknak.

Lehetőségünk van azonban olyan függvényt is megadni, amely diszkontálva tartalmazza az összegeket. Ez azt jelenti, hogy jelenértékben számoljuk a későbbi elvárt jövedelmet.

Ekkor a függvény az alábbi alakot ölti:

$$Hozam_2(s_1, \dots, s_T) = E\left(\left(\frac{1}{1+R}\right)^{i-1} \sum_{i=1}^T r_{I_i}^{s_i}\right),$$

ahol az „R” a piaci kamatlábat jelöli.

Amikor végtelen időhorizonton mozgunk, nem ismert előre az időszakok sorozatának vége, akkor is számolhatunk a hozam<sub>2</sub> függvénnyel, a hozam<sub>1</sub> függvényt pedig nem használhatjuk. Mivel a diszkontálási tényező egynél kisebb, ezért az időszakok végtelenbe tartása esetén az egy fix értékhez konvergál. A hozam<sub>2</sub> függvény másik elnevezése a diszkontált kocánkénti nettó jövedelem. Amennyiben egyenlő hosszú időszakokkal dolgozunk, kiszámíthatjuk az időszakonkénti átlagos nettó jövedelmet is.

$$Hozam_3(s) = \sum_{i=1}^u \pi_i^s \cdot r_i^s,$$

Ahol  $\pi_i^s$  az állapot bekövetkezésének állandósult valószínűsége az s politika mellett. Ugyanezzel a jelöléssel már szerepelt a Markov láncok fogalmai között. Korlátozó tényezők – mint például az alomnagyság – modellbe építése esetén a legalkalmasabb kritérium megadható a következő függvénnyel (például akkor, mikor az egy malacra jutó átlagos elvárt jövedelmet szeretnénk maximalizálni):

$$Hozam_4(s) = \frac{\sum_{i=1}^u \pi_i^s \cdot r_i^s}{\sum_{i=1}^u \pi_i^s \cdot m_i^s},$$

ahol  $m_i^s$  az egy fialásra jutó malacok száma db-ban az i-edik állapotban az s politika mellett. Amikor a hozam<sub>3</sub> függvény alkalmazható, akkor a hozam<sub>4</sub> is alkalmazható, illetve a hozam<sub>4</sub> speciális esete a hozam<sub>3</sub> függvény. Ezek a függvénytípusok Kristensen munkájában találhatók meg (Kristensen, 1996).

### 3.4. Az alkalmazható optimalizációs technikák

#### Érték iteráció

Kiválóan alkalmazható módszer véges időtáv esetén. Az optimális politikát az alábbi függvényegyenlőség rekurzív használata adja meg (Kristensen, 1996):

$$f_i(n) = \max_d \left\{ r_i^d + \beta \sum_{j=1}^u p_{ij}^d f_j(n-1) \right\}, i=1, \dots, u,$$

Ahol a d döntés maximalizálja a jobb oldalát az egyenlőségnek ott van optimum az i állapotban a kérdéses időszakban. Az  $f_i(n)$  a teljes időhorizontjára vonatkozó, várható diszkontált jövedelme,

amely  $i$  állapotban kezdődött és tart  $n$  időszakon keresztül, mielőtt lezárul. Az  $f_i(0)$  egy kiinduló értéke a rendszernek, amikor az az  $i$  állapotban van. Minden időszakban egy optimális politikát választunk a fenti egyenlőség alapján. A hozam<sub>1</sub> célfüggvény használata esetén a  $\beta = 1$  teljesül az egyenlőségben, egyébként pedig  $\beta$  a diszkontálási tényezőt jelenti. Végtelen időtáv esetén az érték iteráció arra használható, hogy megközelítse az optimális politikát. Belátható, hogy a hozam<sub>2</sub> célfüggvény végtelen időtávú változóval történő használata mellett  $\lim_{n \rightarrow \infty} f_i(n) = f_i$ ,  $i=1, \dots, u$ ,

ahol az  $f_i$  fix  $i$ -re konstans érték. A fenti egyenlőség alkalmazása esetén előbb utóbb megfigyelhetjük, hogy az  $f_i(n+1)$  egy idő után majdnem egyenlő lesz  $f_i(n)$ -nel bármely  $i$  indexre. Továbbá ugyanaz a politika adódik számos időszakon keresztül. Ebből már tudhatjuk azt, hogy ez már az optimális politika. Mivel a hozam<sub>3</sub> célfüggvény egy speciális esete a hozam<sub>4</sub> célfüggvénynek, ezért a kritériumot megadhatjuk csak a hozam<sub>4</sub> célfüggvénnyel is. Ebben az esetben az  $f_i(n)$  a várható jövedelmet jelöli, amikor a folyamat egy kezdő időszak  $i$  állapotától addig tart, amíg  $n$  egység fizikai mennyiséget elő nem állítottunk. Az optimális politika ezen  $n$  egységnyi output előállításához megadható az alábbi rekurzív formulával (Kristensen, 1996):

$$f_i(n) = \max_d \left\{ a \left( \frac{nr_i^d}{m_i^d} + f_i(0) \right) + (1-a) \left( r_i^d + \sum_{j=1}^u p_{ij}^d f_j(n - m_i^d) \right) \right\}, n=1, \dots$$

$$\text{Ahol } a = \begin{cases} 1 & m_i^d \geq n \\ 0 & m_i^d < n \end{cases}$$

Minden azon a feltételezésen alapszik, hogy az  $\frac{r_i^d}{m_i^d}$  mutató (jövedelem/kibocsátás) konstans marad az

egész időszakban. Ha  $n$  értéke elég nagy, akkor  $a=0$  adódik. Howard (1971) tanulmányában az  $m_i^d$  az időszak várható hosszaként értelmezett paraméter. Így a hozam<sub>4</sub> függvény az időszakonkénti elvárt jövedelemként fogható fel.

### **Politika iteráció**

A politika iterációt végtelen időtáv esetében használhatjuk. Ellentétben az érték iterációval, a politika iteráció mindig optimális megoldást szolgáltat. Kombinálható mind a hozam<sub>2</sub> függvény végtelen időtávú változatával, mind a hozam<sub>3</sub> és a hozam<sub>4</sub> függvénnyel. Az  $f_i^s$  a várható jövőbeli jövedelme a folyamatnak, ha az  $s$  politikát követjük és a Hozam<sub>2</sub> végtelenített változatát alkalmazzuk. A Hozam<sub>3</sub> függvény és a Hozam<sub>4</sub> függvény alkalmazásával az  $f_i^s$  az  $i$  állapot relatív hasznosságát jelenti  $s$  politika mellett.

Az  $f_i^s$  a várható jövőbeli jövedelme a folyamatnak, ha az  $s$  politikát követjük és a Hozam<sub>2</sub> végtelenített változatát alkalmazzuk. A Hozam<sub>3</sub> függvény és a Hozam<sub>4</sub> függvény alkalmazásával az  $f_i^s$  az  $i$  állapot relatív hasznosságát jelenti  $s$  politika mellett.

### **3.5. Hierarchikus Markov folyamatok**

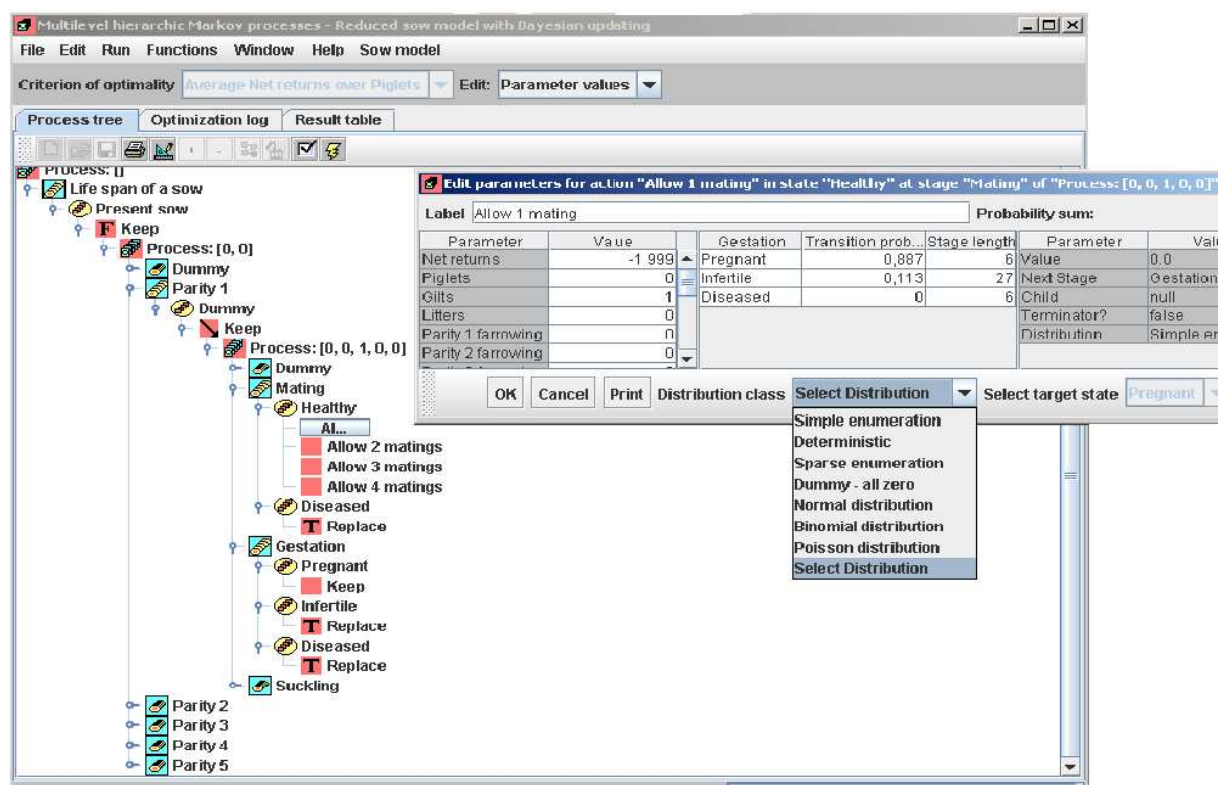
Kristensen dán professor a Markov döntési folyamatok alapján fejlesztette ki a Hierarchikus Markov folyamatok (HMP) technikáját, amely lehetővé teszi, hogy nagy állapotterű (akár 6,8 millió állapotból álló) rendszerekkel dolgozzunk (Houben et al., 1994; Kristensen, 2004a). A HMP alkalmazása ugyanakkor nem zárja ki az érték iteráció és a politika iteráció alkalmazását.

A Markov döntési folyamatok sorozatát részfolyamatnak nevezzük a HMP-n belül, amelyet szintén egy Markov döntési folyamatba ágyaztunk be, amelyet pedig a alapfolyamatnak nevezünk. Az alapfolyamat lehetséges állapotai meghatározzák a részfolyamatok állapotait. Az alapfolyamat

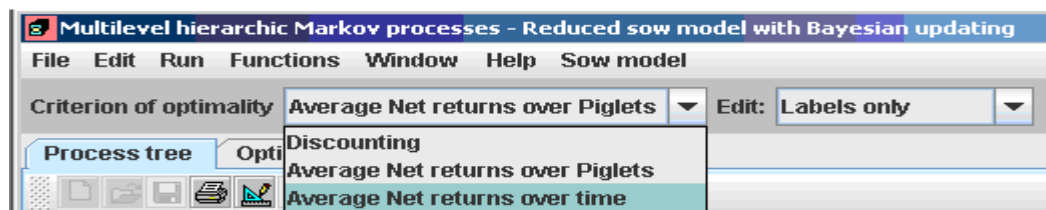
állapotai adott tulajdonságaira vonatkoznak, amelyek az állatok között változnak, de ugyanazon állatra vonatkozóan konstansnak tekinthetők a vizsgálat ideje alatt (pl. genetikai helyzet, tartásmód, stb.). Azok a tényezők, amelyek az idő függvényében változhatnak, a részfolyamatok állapotváltozóiként szerepelnek a modellben. Minden részfolyamat véges számú időszakokból (stages) áll (ezek lehetnek például az állat életének különböző életszakaszai). Az alapfolyamat állapotaiból származó hozamot a részfolyamatokból származó hozam határozza meg. A modellezett szituációtól függően a HMP úgy határozza meg az optimális döntési stratégiát, hogy egy megfelelő, előre definiált függvényt maximalizál. Lehetőség van a fialásonkénti átlagos jövedelem, vagy az átlagos jövedelem/malac vagy az átlagos elvárt teljes diszkontált jövedelem kiszámítására is.

## 4. Eredmények

A HMP-ket modellező programot Kristensen dán professzor készítette el (Kristensen, 1996). A program egy saját fejlesztőfelületet kínál fel (1. ábra), amelyben tetszőlegesen szerkeszthetünk folyamatokat, alfolyamatokat, állapotokat, de beépített modellek is rendelkezésünkre állnak. A tanulmányunkban egy ilyen beépített modellt (az ún. koca modell) ismertetünk, amelyek állatok életét szimulálják. A kocák esetén a megoldandó probléma a termékenyítés kérdése. Ha termékenyítettünk egy állatot, akkor az állat a következő állapotokba kerülhet: az állat vemhes lehet, meddő maradhat vagy megbetegedhet (1. ábra). A 3 állapotnak megfelelően döntünk a sorsáról, és minden egyes döntés következtében a ciklus a következő időszakban a megfelelő állapotba adott valószínűséggel kerül. Ezen valószínűségek becslése egyéb eljárásokkal az alapadatok alapján történik. Például ha 1-szer termékenyítjük az állatot, akkor 0,887 valószínűséggel vemhesül a következő időszak alatt (1. ábra). A valószínűségeket előre definiált eloszlások szerint is megadhatjuk. Egyéb értékeket is be kell állítanunk az optimalizáció sikere érdekében, például a várt nettó jövedelmet az adott döntés esetén, született malacok száma, kocasüldők száma, alomszám, fialások száma stb...).



1. ábra. A kocák termékenyítésének modellje



**2. ábra.** Az alkalmazható optimalizálási kritériumok

A 2. ábra az alkalmazható hozam függvényeket, vagyis az optimalizálás kritériumait tünteti fel. A discounting a már bemutatott Hozam2 függvény, a másik két kritérium a Hozam3 és Hozam4 függvény.

Level	Founder state	Founder action	Level 1 stage	Level 1 state	Level 1 action	Level 2 stage	Level 2 state	Level 2 action	Value
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Replace	62 736,355
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Keep	63 508,512
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Replace	62 952,688
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Keep	63 740,094
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Replace	63 169,535
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Keep	63 975,039
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Replace	63 386,891
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Keep	64 217,02
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Replace	63 604,754
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Keep	64 435,398
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Replace	63 823,133
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Diseased	Replace	59 738,316
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Mating	Healthy	Allow 1 mati...	60 789,465
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Mating	Healthy	Allow 2 mati...	60 876,891
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Mating	Healthy	Allow 3 mati...	60 885,59
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Mating	Healthy	Allow 4 mati...	60 886,465
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Mating	Diseased	Replace	59 738,316
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Gestation	Pregnant	Keep	61 287,316
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Gestation	Infertile	Replace	60 064,656
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Gestation	Diseased	Replace	59 738,316
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Keep	60 124,84

**3. ábra.** Az optimális döntések sorozata és az elérhető jövedelem

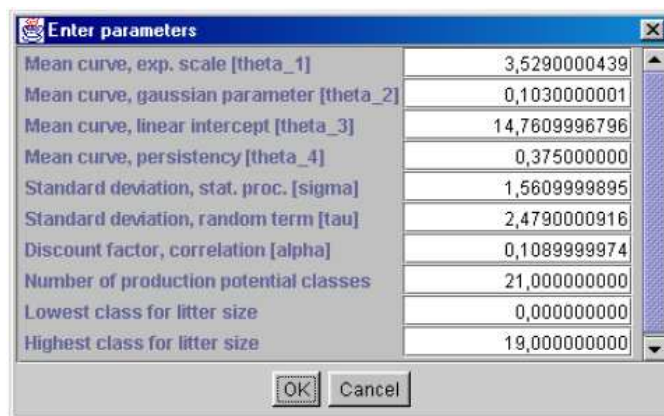
Az optimalizálás eredménye a 3. ábrán látható, a program megadja az alkalmazandó stratégiát (tartjuk-e meg az állatot, vagy ne, hány termékenyítésben vegyen részt, stb.), és a kapcsolódó relatív hasznosságokat, ha adott döntést hozunk szemben egy másikkal.

A következőkben bemutatjuk a „Koca modell” paramétereinek beállítási lehetőségeit és a szimuláció eredményeit. Az alapparaméterek az eredeti biológiai modellből származnak (Kristensen, 2004b).



**4. ábra.** A koca modell paramétereinek beállítása

A 4. ábra mutatja, hogy milyen paramétereket kell beállítani az optimalizálás előtt. Elsőként a várható fialások számát kell megadni. Majd az egy fialásra jutó termékenyítések minimális és maximális értékét állítjuk be. Meg kell adni a felhasznált kanok számát is és a szoptatás idejét hetekben.



**5. ábra.** Az alomnagyságát meghatározó modell paramétereinek beállítása

Az 5. ábrán az első négy paraméter a különböző eloszlások átlagait határozza meg, míg a következő kettő a szórásokat definiálja. A 7. sorban az összefüggés szorosságát lehet beállítani. Az utolsó három sor a technikai környezet és a modell szerkezetének változtatását teszi lehetővé.



Enter parameters	
Sow weight, parity 1	157.00000000
Sow weight, parity 2	161.00000000
Sow weight, parity 3	205.00000000
Sow weight, parity 4	219.00000000
Sow weight, parity 5	236.00000000
Sow weight, parity 6	252.00000000
Sow weight, parity 7	259.00000000
Sow weight, parity 8	267.00000000
Sow weight, parity 9	270.00000000
Sow weight, parity 10	270.00000000
Sow weight, parity 11	270.00000000
Sow weight, parity 12	270.00000000
Sow weight, parity 13	270.00000000
Sow weight, parity 14	270.00000000
Sow weight, parity 15	270.00000000
Sow weight, parity 16	270.00000000
Drop-out, parity 0, constant [beta_0]	-3.720200618
Drop-out, parity 0, litter size effect [beta_1]	0.00000000
Drop-out, parity 0, square of litter size [beta_2]	0.00000000
Drop-out, parity 1, constant [beta_0]	-2.648499957
Drop-out, parity 1, litter size effect [beta_1]	0.00000000
Drop-out, parity 1, square of litter size [beta_2]	0.00000000
Drop-out, parity 2, constant [beta_0]	-0.337199988
Drop-out, parity 2, litter size effect [beta_1]	-0.366499902
Drop-out, parity 2, square of litter size [beta_2]	0.013700000
Drop-out, parity 3, constant [beta_0]	2.620600183
Drop-out, parity 3, litter size effect [beta_1]	-0.533399991
Drop-out, parity 3, square of litter size [beta_2]	0.013700000
Drop-out, parity 4, constant [beta_0]	2.620600183
Drop-out, parity 4, litter size effect [beta_1]	-0.533399991

OK Cancel

6. ábra. A koca paramétereinek beállítása

A 6. ábrán a koca különböző jellemzőinek a beállítását mutatjuk be. Az adatok első csoportja a koca testtömegének az alakulását mutatja a különböző fialások esetében egy standard görbe alapján. A többi adatnál a különböző fialásokhoz adhatók meg beállítási paraméterek (konstans, alomnagyság hatása, alomnagyság hatás négyzete), amelyek szükségesek az alkalmazott függvények kiszámításához a nem tervezett selejtezések esetében. A modellezés során valójában sok fialási paramétert összevon a modell és azokkal kalkulálja a végeredményt.

Enter parameters	
Daily feed intake, mating period	3.75000000
Daily feed intake, pregnancy, week 1-4	2.400000954
Daily feed intake, pregnancy, week 5-12	2.400000954
Daily feed intake, pregnancy, week 13-16	3.50000000
Daily feed intake, pregnancy, last days	2.25000000
Piglet feed intake per piglet, 3rd week	0.029999993
Piglet feed intake per piglet, 4th week	0.119999973
Piglet feed intake per piglet, 5th week	0.600000238
Ad libitum feed intake, constant term [c0]	2.191999123
Ad libitum feed intake, regression on parity [c1]	0.296999909
Ad libitum feed intake, regression on squared parity [c2]	-0.021999999
Ad libitum feed intake, regression on piglets [c3]	0.224000069
Ad libitum feed intake, regression on squared piglets [c4]	-0.008000004
Ad libitum feed intake, regression on weight gain [c5]	-0.007000002
Ad libitum feed intake, regression on weaning age [c6]	0.014999997

OK Cancel

7. ábra. A takarmány fogyasztás paramétereinek beállítása

A 7. ábrán a vemhesség és a szoptatás alatti takarmány fogyasztás paramétereinek beállítását mutatjuk be. Külön lehet megadni a termékenyítés idejére és a vemhesség 4 eltérő időszakára a napi takarmány fogyasztás adatait. A malacok 3. – 5. heti takarmány felvételét is be tudjuk állítani. Az ad-libitum takarmányozás regressziós paramétereit is itt tudjuk definiálni  $c_0 - c_6$ -ig (a fialások számát, a fialások számának négyzetét, a malacszámot, a malacok számának négyzetét, a tömeggyarapodást, a választási kort).

Enter parameters	
Feed price, mating period	1,2999999523
Feed price, gestation period	1,2999999523
Feed price, suckling period	1,2999999523
Feed price, piglet feed	2,7999999523
Basic price per piglet sold	249,0000000000
Price of replacement gilt	1704,0000000000
Price per kg, old sow	6,0100002289
Reduction factor, diseased sows	0,6999999881

OK Cancel

**8. ábra.** A különböző árak rögzítése

A 8. ábra az árak paramétereinek beállítását mutatja. A koca takarmányának árai megadhatók a különböző időszakokra (termékenyítés, vemhesség, szoptatás) és a malactáp ára is itt rögzíthető. Beállítható a malacár, a tenyészsüldő ára, a selejtkoca ár és a betegkoca csökkentett értéke.

Enter parameters	
Basic conception rate	0,9169999957
Fixed price of mating	0,0000000000
Relative parity effect of conception rate, parity 1	0,9670000076
Relative parity effect of conception rate, parity 2	0,9739999771
Relative parity effect of conception rate, parity 3	0,9909999967
Relative parity effect of conception rate, parity 4	1,0000000000
Relative parity effect of conception rate, parity 5	0,9900000129
Relative parity effect of conception rate, parity 6	0,9710000157
Relative parity effect of conception rate, parity 7	0,9860000014
Relative parity effect of conception rate, parity 8	0,9200000167
Relative parity effect of conception rate, parity 9	0,8719999790
Relative parity effect of conception rate, parity 10	0,8719999790
Relative parity effect of conception rate, parity 11	0,8719999790
Relative parity effect of conception rate, parity 12	0,8719999790
Relative parity effect of conception rate, parity 13	0,8719999790
Relative parity effect of conception rate, parity 14	0,8719999790
Relative parity effect of conception rate, parity 15	0,8719999790
Relative parity effect of conception rate, parity 16	0,8719999790
Effect of remating 0 on conception rate	-0,0000000000
Effect of remating 1 on conception rate	-0,0299999993
Effect of remating 2 on conception rate	-0,0599999987
Effect of remating 3 on conception rate	-0,0900000036
Effect of remating 4 on conception rate	-0,1199999973

OK Cancel

**9. ábra.** A termékenyítés paramétereinek beállítása

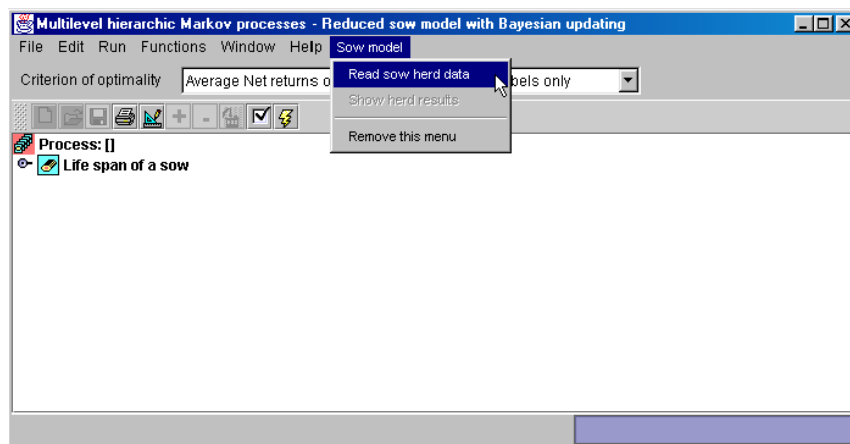
A 9. ábrán a termékenyítés paramétereit mutatjuk be. Minden fialásnál külön – külön megadhatók az értékek. A termékenyítési arányt az alaparánytól és a fialástól függő relatívhatásként határozzuk meg. A termékenyítés költségénél változatlan árral kalkulálunk. A tényleges termékenyítési arányt úgy számoljuk ki az  $n$ -ik fialásnál, hogy az alaptermékenyítési rátát megszorozzuk az  $n$ -ik fialás relatív hatásával. A visszaivarzó állatok ismételt termékenyítésének költségeit is figyelembe tudjuk venni 4 újratermékenyítésig.

Enter parameters	
Piglet mortality, first parity	0,1679999977
Piglet mortality, other parities	0,2329999954

OK Cancel

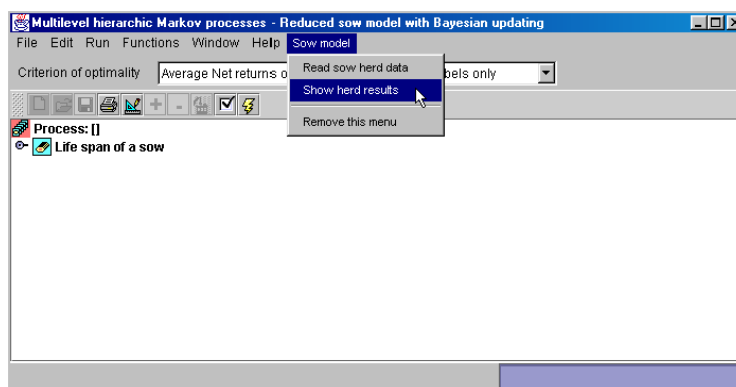
**10. ábra.** A malacok elhullásának alakulása az első és a későbbi fialások során

A 10. ábrán bemutatjuk a malacok elhullásának alakulását az első és a későbbi fialások során.



11. ábra. A „Koca modell” főmenüje.

A 11. ábrán látható a „Koca modell” főmenü, amelyből egy már létező koca állomány adatait nyithatjuk meg a programban. Így tudjuk gyorsan beolvastatni a már meglévő adatainkat.



12. ábra. A „Koca modell” főmenüje az állomány eredményeivel

A „Koca modell” főmenüből lehetőség van arra is, hogy egy már korábbi vizsgálat elmentett eredményeit megnyissuk (12. ábra). Így lehetővé válik a már megoldott eredmények szakmai vizsgálata.

Multilevel hierarchic Markov processes - Reduced sow model with Bayesian updating

File

Edit

Run

Functions

Window

Help

Sow model

Criterion of optimality

Average Net returns over time

Edit:

Labels only

Sow number	Parity	Previous litter size potential	Litter size	Updated litter size potential	Retention pay-off	Max. number of matings
126	7	6	11	-1	22,798	1
156	7	3	14	3	44,015	1
163	7	9	17	8	98,858	2
167	7	-2	13	1	29,647	1
172	8	2	11	-2	14,292	1
176	7	-4	12	-1	15,72	1
194	8	3	14	3	57,905	2
196	8	-1	7	-7	-33,992	1
198	8	4	14	3	57,905	2
200	8	5	19	8	120,507	2
202	8	2	14	3	57,905	2
203	8	7	13	2	48,873	2

13. ábra. Az optimális döntések sorozata és az elérhető jövedelem az állomány minden kocájára kiszámítva

A 13. ábrán látható, hogy a szimulált telep összes kocájára érték iterációval kiszámítottuk az elérhető jövedelmet és a kocánkénti döntések sorozatát. Minden egyes kocának meg van adva az

azonosítója, a fialásainak a száma, az előző lehetséges alomszámai, a legutolsó alom mérete, a potenciális alomszám értéke úgy, hogy figyelembe vettük a legutolsó alom nagyságát, a koca termelésben tartásának jövőbeli jövedelmezősége, a vemhesítések számának maximuma mielőtt a kocát termékenységi zavarok miatt selejtezni kellene. A legérdekesebb információ az elérhető jövedelem, amelyet tekinthetünk egy féle gazdasági indexnek, ami a koca sokféle termelési tulajdonságából alakult ki. A negatív érték azt jelenti, hogy célszerű lenne a koca selejtezése. A pozitív szám azt jelzi, hogy a kocát érdemes a termelésben tartani. A numerikus érték azt mutatja, hogy a költségek eltérnek az optimális értékektől.

## Irodalomjegyzék

- Barbour, A. 2004. Bioinformatik II, Probability and Statistic, Markov chains, <http://www.math.unizh.ch/~schumi/bioinf2.html>
- Bellman, R. E. 1957. Dynamic Programming. Princeton: Princeton University Press.
- Bellman, R. E. 1961. Adaptive control process: a guide tour. Princeton: Princeton University Press.
- Bellman, R. E. and Dreyfus, S. E. 1962. Applied dynamic programming. Princeton: Princeton University Press.
- Bellman, R. E. and Kalaba, R. 1965. Dynamic programming and modern control theory. New York: Academic Press.
- Hadley, G. (1964): Nonlinear and dynamic programming. Reading, Massachusetts: Addison-Wesley.
- Houben, E. H. P., R. B. M. Huirne, A. A. Dijkhuizen, and A. R. Kristensen. 1994. Optimal replacement of mastitis cows determined by a hierarchic Markov process. Journal of Dairy Science 77: 2975-2993.
- Howard, R. A. 1960. Dynamic programming and Markov Process. Cambridge, Massachusetts: The M.I.T. Press.
- Howard, R.A. 1971. Dynamic probabilistic systems. Volume II: Semi-Markov and decision process. New York: John Wiley & Sons, Inc.
- Jewell, W. 1963. Markov renewal programming I and II. Operations Research 11: 938-971.
- Kennedy, J. O. S. 1981. Applications of dynamic programming to agriculture, forestry and fisheries: Review and prognosis. Review of Marketing and Agricultural Economics 49: 141-173.
- Kristensen, A. R. 2004a. A sow replacement model using Bayesian up-dating in a 3-level Hierarchic Markov process: II. Optimization model. Livestock Production Science 87(1): 25-36.
- Kristensen, A. R. 2004b. A sow replacement model using Bayesian up-dating in a 3-level Hierarchic Markov process: I. Biological model. Livestock Production Science 87(1): 13-24.
- Kristensen, A. R., 1996. Herd management: Dynamic programming/Markov decision processes, Dina Notat No. 49.
- Kristensen, A.R. 1991. Maximization of net revenue per unit of physical output in Markov decision processes. European Review of Agricultural Economics 18: 231-244.
- Kristensen, A.R. and Jorgensen, E. 1995. Applicational perspectives of recent developments in dynamic programming methods for herd management support. Dina Notat No. 33.
- Ross, S. M. 1970. Applied probability models with optimization applications. San Francisco, California: Holden-Day.
- Wal, J., and Wessels, J. 1985. Markov Decision Processes. Statistica Neerlandica 39(2): 219-233.
- White, C.C., and White, D.J. 1989. Markov decision process. European Journal of Operational Research 39: 1-16.