Magyar Agrárinformatikai Szövetség
Hungarian Association of Agricultural Informatics

Agrárinformatika Folyóirat. 2012. 3. évfolyam 2. szám
Journal of Agricultural Informatics. 2012 Vol. 3, No. 2

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[1], Balogh Péter[2]

I N F O
Received 13 Oct 2012
Accepted 09 Dec 2012
Available on-line 28 Dec 2012
Responsible Editor: Rajkai, K.

Kulcsszavak:
Markov processes, sow model, mating, gestation, net revenue

A B S T R A C T

In this study we discuss the Markovian chain-based decision processes and their developed variant called Hierarchic Markovian Processes. The optimizing possibilities of such processes are presented in detail. Moreover, we introduce a free available software based on these processes and developed by Danish researchers for supporting decisions in animal breeding. Among the several models the reduced sow model (with gestation) were chosen for presentation. We describe the basic settings and parameters for running the software as well as we calculate the average net return over time and the series of decisions per sow in case of simulated sow herd data by applying the value iteration technique. We also present the results of decisions on keeping an animal in production as well as on determining the number of matings of a sow. We also give examples of the development of the relative utility values related to such decisions.

I N F O
Beérkezés 2012. Okt. 13.
Elfogadás 2012. Dec. 09.
On-line elérés 2012. Dec. 28.
Felelős szerkesztő: Rajkai K

Kulcsszavak:
Markov folyamatok, koca modell, termékenyítés, vemhesség, nettó jövedelem

Ö S S Z E F O G L A L Ó

Jelen tanulmányban a Markov láncokon alapuló Markov döntési folyamatokat tárgyaljuk, valamint ezek továbbfejlesztett változatát, a Hierarchikus Markov Folyamatokat. Részletesen leírjuk ezen folyamatok optimalizációs lehetőségeit. Ismertetünk továbbá egy dán kutatók által állattenyésztési döntések támogatására kifejlesztett, e folyamatokon alapuló és szabadon letölthető programot. A program számos lehetőségéből a kocanevelés és vemhesítés modell használatát mutatjuk be. Megadjuk a program futtatásához szükséges alapparamétereket, beállításokat, valamint egy modellezett állattartó telep kocáira érték iterációval számítjuk az elérhető jövedelmet és a kocánkénti döntések sorozatát. Bemutatjuk az állat termelésben tartására, illetve a koca termékenyítési számára hozott döntések eredményét. Példán mutatjuk be továbbá az ilyen döntésekhez kapcsolt relatív hasznosság alakulását.

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 alkalmazását a problémák megoldásában. A mezőgazdaságban számos matematikai módszert alkalmaztak már, mint például a hálótervezést, a lineáris programozást, a függvény analízist és a matematikai statisztikát. Ezek az analitikus módszerek azonban több korlátot is hordoznak. A lineáris programozás esetében nagy kötöttséget jelent a lineáris összefüggések keresése, ami leszűkíti az alkalmazási területet. A járhatóbb út az úgynevezett szimulációs kísérlettel történő modellezés, amely nagyobb rugalmasságú és szélesebb a felhasználási területe. A szimulációs módszer nem

[1] Debreceni Egyetem, kovacss@agr.unideb.hu
[2] Debreceni Egyetem, baloghp@agr.unieb.hu

optimalizációs eljárás, de az általunk bemutatandó módszer a dinamikus programozásból ered és beépített optimalizációs modulja is van. Jelen tanulmányban bemutatjuk a Hierarchikus Markov Folyamatok módszertanát, optimalizációs lehetőségeit. Ismertetünk egy dán kutatók által állattenyésztési döntések támogatására kifejlesztett, ingyenesen letölthető szoftvert. A program alkalmazásának lehetőségei közül a kocák vemhesítésének problémáját tárgyaljuk. A módszertan bemutatásában a felhasznált fogalmakra támaszkodunk.

2. Történeti áttekintés

A dinamikus programozás módszertanát először Bellman publikálta (Bellman,1957). Könyvében a szekvenciális döntési problémák megoldására új numerikus módszert adott közre, melynek elemei a „Bellman-féle optimalitási követelmények" és a funkcionális egyenletek. Első művét követően még több könyve jelent meg (Bellman, 1961; Bellman and Dreyfus, 1962; Bellman és Kalaba, 1965). A dinamikus programozást sokan triviális számítási eszköznek tartották, mások azonban többre értékelték. Később igazolódott, hogy sem nem triviális, sem nem kellően általános a problémák megoldásában. A dinamikus programozás fő ismérve a szekvenciális közelítés. Ez adja a sorozatos döntésekre történő alkalmazhatóságát. A sorozatos döntési probléma egyik szemléletes példája az állattenyésztésben az állat „selejtezése", illetve az új kocasüldő beállítása, az inszeminálás, a vemhesítés és az orvosi ellátás. A módszer azért is releváns az állattenyésztésben, mert az állat jellemzői (pl.: alacsony vagy magas hozam, termékenyülési képesség, alomszám, stb.) valószínűségekkel becsülhetők, és a döntések az aktuális megfigyelésekre vonatkoznak. A dinamikus programozás Markov-folyamatokkal történő összekapcsolását Howard dolgozta ki (Howard, 1960). A két módszer kombinálásával jött létre a Markov döntési folyamat (MDF), melyet Howard vezetett be. Ő ismertette a politikai iterációt, és az érték-iterációs technikát is, amellyel az MDF optimalizálható.

A politikai iterációt két optimalitási kritérium miatt fejlesztették ki. Nevezetesen a teljes időhorizontra vonatkozó, elvárt diszkontált jövedelem és az elvárt átlagos időszaki jövedelem maximalizálására. Jewell mutatta be a politika iterációt az átlagos jövedelem maximalizálására a teljes időhorizonton, és azt semi-Markov döntési folyamatnak nevezte (Jewell, 1963). Ebben az MDF-ben az időszakok hossza véletlen módon változik. Howard az érték iterációra is kidolgozta a semi-Markov döntési folyamatokat (Howard, 1971). Megjegyezzük, hogy a lineáris programozás módszerét már az említett két optimalizációs technika előtt alkalmasnak tartották az MDF-ek optimalizálására (Hadley, 1964). Azonban a kutatók ezt a technikát egyik állattenyésztési modellben sem alkalmazták, 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 kezeléssel és az inszeminálással. Az eddigi módszertani fejlesztések nagy részét Kristensen és Jorgensen (1995) publikálta. Az alkalmazások 3 fő nehézsége a változékonyság, a reproduktív ciklusok és a korlátok (például a malacnevelő képesség határai, vagy a véges elhelyezési kapacitás). Az első két nehézséget az MDF kezeli, az utóbbi pedig nem változtatható.

3. Anyag és módszer

3.1. Markov lánc

A Markov döntési folyamatok alapja a Markov-lánc. Megértésükhöz szükség van néhány alapvető fogalomra a Markov-lánc elméletből. A Markov-lánc a sztochasztikus folyamatok speciális típusa. Diszkrét idejű sztochasztikus folyamatot akkor nevezünk Markov-láncnak, amikor a $P(X_{t+1} = i_{t+1} \mid X_t = i_t)$ feltételes valószínűség fennáll.

Ez azt jelenti, hogy a t+1-edik időszak valószínűség eloszlása a t-edik időszak állapotától függ, de független a lánc korábbi időszakaiétól, amelyeken a folyamat áthaladt. Ebből adódóan feltehetjük, hogy minden i és j állapotra minden t időszakban fennáll a t-től függő $P(X_{t+1} = j \mid X_t = i) = p_{ij}$ és $P(X_{t+1} = j \mid X_t = i)$ valószínűség. A p_{ij} jelöli annak a valószínűségét, hogy az adott rendszer a t

időszakbeli i-edik állapot után a t+1-edik időszak j-edik állapotába kerül. Innen ered *p*-re az átmenet valószínűség elnevezés. Továbbá azok a valószínűségek, amelyek az első időszak i-edik állapotának bekövetkezésére vonatkoznak, azok a kezdő valószínűség eloszlások, jelük q_i. Így $P(X_1 = i) = q_1$.
Végül minden t időszakbeli i állapotra teljesül a következő egyenlet:

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

Ebből az is következik, hogy az átmenet valószínűségek alkotta mátrix elemei nem negatívak, és minden sorösszeg 1-et ad. A p_{ij} valószínűségeket gyakran csak egylépéses valószínűségeknek nevezik, mivel a t időszakból t+1-edik időszakra térünk át. Szokásos erre a $p_{ij}(1)$ jelölést használni. Minden Markov-lánccal megoldható problémában az a kérdés, 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 stacionaritást használjuk ki, ami azt jelenti, hogy a valószínűségek függetlenek t időszaktól. Ebből adódik, hogy:

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

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 $\pi = (\pi_1, \pi_2, ..., \pi_n)$ vektor (Barbour, 2004):

$$\lim_{n\to\infty} P^n = \begin{bmatrix} \pi_1 & \pi_2 & . & . & \pi_s \\ \pi_1 & \pi_2 & . & . & \pi_s \\ . & . & . & . & . \\ . & . & . & . & . \\ \pi_1 & \pi_2 & . & . & \pi_s \end{bmatrix}$$

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

Ez azt jelenti, hogy a Markov-lánc π_j valószínűséggel állandósul a j állapotban, ami független a kiinduló állapottól. A π 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 π 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 elvárt 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 visszafele haladva maximalizálni kell a funkcionális függvény jobboldalát.
Legyen adott egy diszkrét Markov döntési folyamat egy véges állapottérrel, U={állapot i=1,2,…,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 időszaknak nevezzük. Néhány modell az $m_i^{\ d}$ fizikai mennyiséget (egy anyakocára jutó alomnagyság, vagy életteljesítmény) 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:

$$Hozam_1(s_i,...,s_T) = E(\sum_{i=1}^{T} r_{I_i}^{\ s_i}),$$

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:

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

ahol az „R” a piaci kamatlábat jelöli.

Amikor végtelen időhorizonton mozgunk, nem ismert előre az időszakok sorozatának vége, akkor is számolhatunk a $hozam_2$ függvénnyel, a $hozam_1$ függvényt pedig nem használhatjuk. Mivel a diszkontálási tényező egynél kisebb, ezért az időszakok végtelenbe tartása esetén az egy fix értékhez konvergál. A $hozam_2$ 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):

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

ahol $m_i^{\ s}$ az egy fialásra jutó malacok száma db-ban az i-edik állapotban az s politika mellett. Amikor a $hozam_3$ függvény alkalmazható, akkor a $hozam_4$ is alkalmazható, illetve a $hozam_4$ speciális esete a $hozam_3$ függvény. Ezek a függvénytípusok Kristensen munkájában találhatók meg (Kristensen, 1996).

3.4. Az alkalmazott optimalizációs technikák

Érték iteráció

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

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

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 $hozam_1$ célfüggvény használata esetén a $\beta = 1$ teljesül az egyenlőségben, egyébként pedig β a diszkontálási tényezőt jelenti. Végtelen időtáv esetén az érték iteráció arra használható, hogy megközelítse az optimális politikát. Belátható, hogy a $hozam_2$ célfüggvény végtelen időtávú változóval történő használata mellett $\lim_{n \to \infty} f_i(n) = f_i$, i=1,…,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 $hozam_3$ célfüggvény egy speciális esete a $hozam_4$ célfüggvénynek, ezért a kritériumot megadhatjuk csak a $hozam_4$ 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):

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

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

Minden azon a feltételezésen alapszik, hogy az $\dfrac{r_i^{\ d}}{m_i^{\ d}}$ 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 $hozam_4$ 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 $hozam_2$ függvény végtelen időtávos változatával, mind a $hozam_3$ és a $hozam_4$ függvénnyel. Az $f_i^{\ s}$ a várható jövőbeli jövedelme a folyamatnak, ha az s politikát követjük és a $Hozam_2$ végtelenített változatát alkalmazzuk. A $Hozam_3$ függvény és a $Hozam_4$ 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 $Hozam_2$ végtelenített változatát alkalmazzuk. A $Hozam_3$ függvény és a $Hozam_4$ függvény alkalmazásával az $f_i^{\ s}$ az i állapot relatív hasznosságát jelenti s politika mellett.

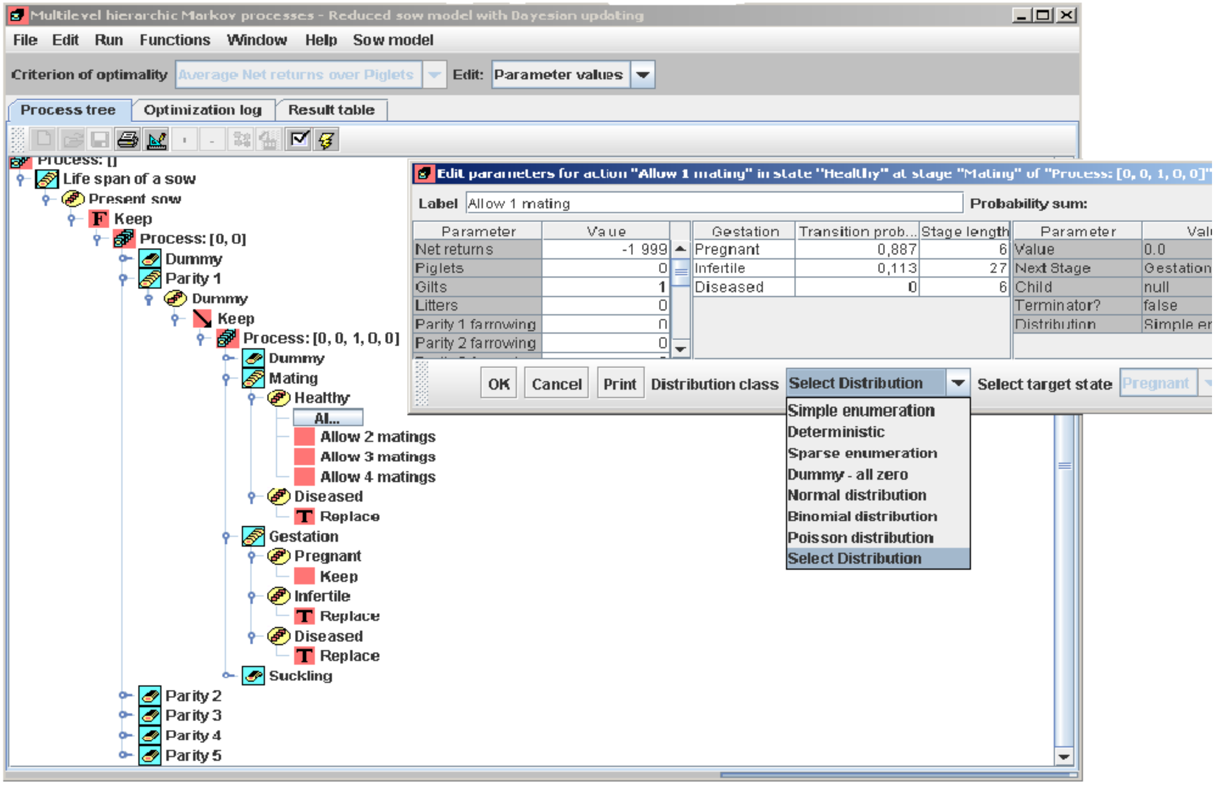
3.5. Hierarchikus Markov folyamatok

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

A Markov döntési folyamatok sorozatát részfolyamatnak nevezzük a HMF-en 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 tekinthetőek 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 HMF 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.

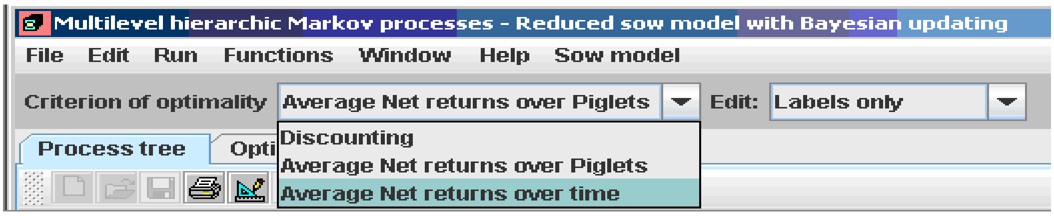
4. Eredmények

A HMF-eket 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 un. 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 időszakasz 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…).



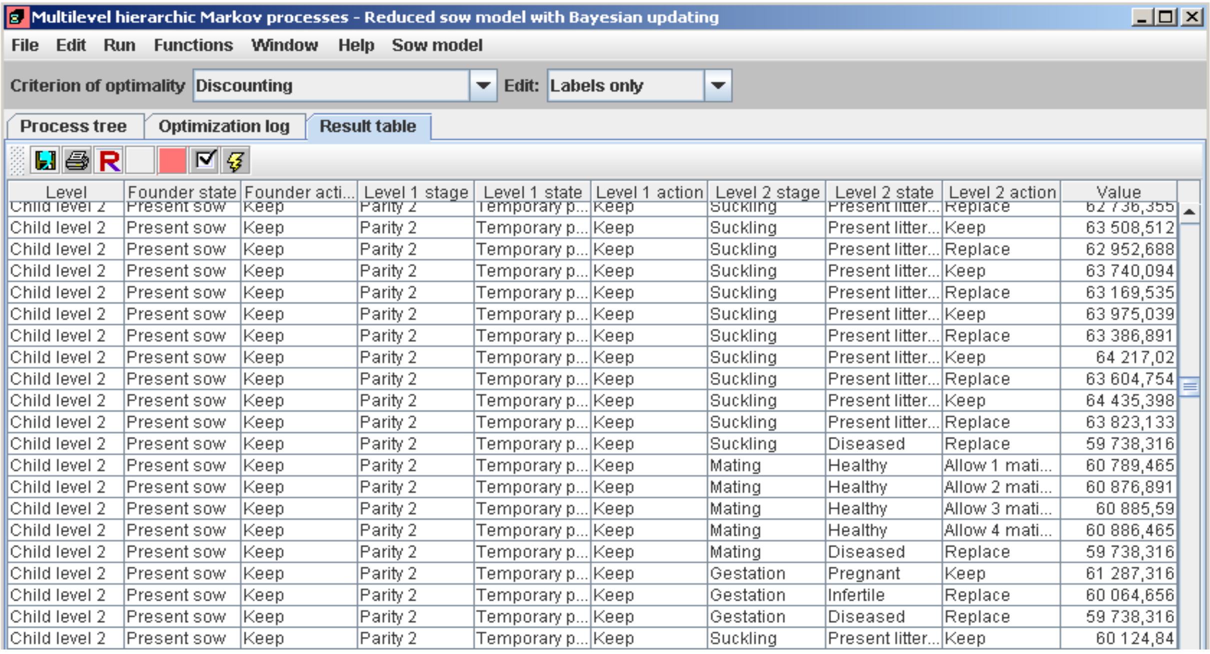
(Mating: termékenyítés; Gestation: vemhesség; Suckling: szoptatás; Infertile: meddő; Diseased: megbetegedett; Pregnant: vemhes; Parity: fialások száma; *Farrowing:* fialás; *Litters*: almonagyság; *Gilts:* kocasüldők száma; *Piglets:* malacszám, *Net returns:* Nettó jövedelem; Select Distribution: eloszlás választó (normális, Binomiális, Poisson, 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 Hozam4 (Average Net returns over Piglets) és Hozam3 (Average Net returns over time) függvény.



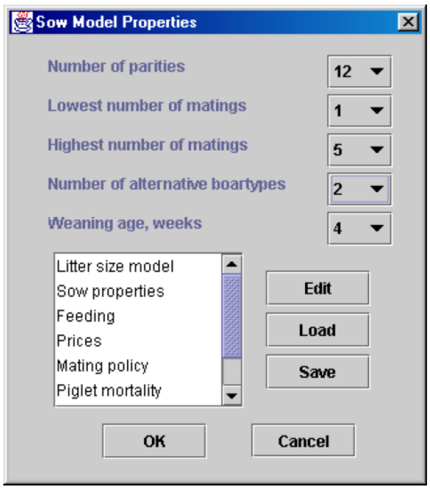
Level	Founder state	Founder acti...	Level 1 stage	Level 1 state	Level 1 action	Level 2 stage	Level 2 state	Level 2 action	Value
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	Present litter...	Replace	62 736,355
Child level 2	Present sow	Keep	Parity 2	Temporary p...	Keep	Suckling	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

* (az elnevezések jelentését lásd az 1. ábra alatt)

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

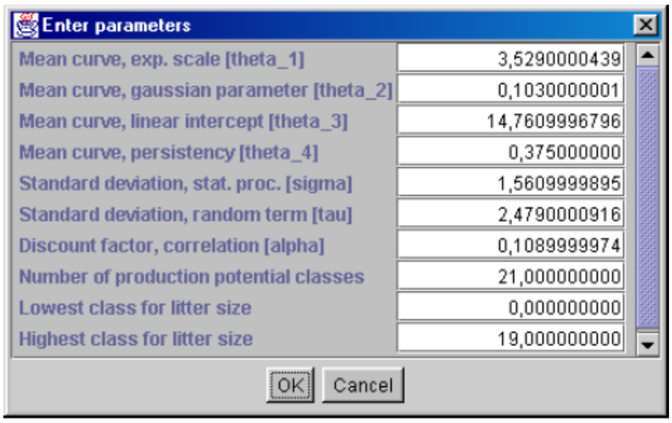
Az optimalizálás eredménye a 3. ábrán látható, a program megadja az alkalmazandó stratégiát (tartsuk-e meg az állatot, vagy ne, hány termékenyítésben vegyen részt, stb..), és a kapcsolódó relatív hasznosságokat, ha adott döntést hozunk szemben egy másikkal.

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



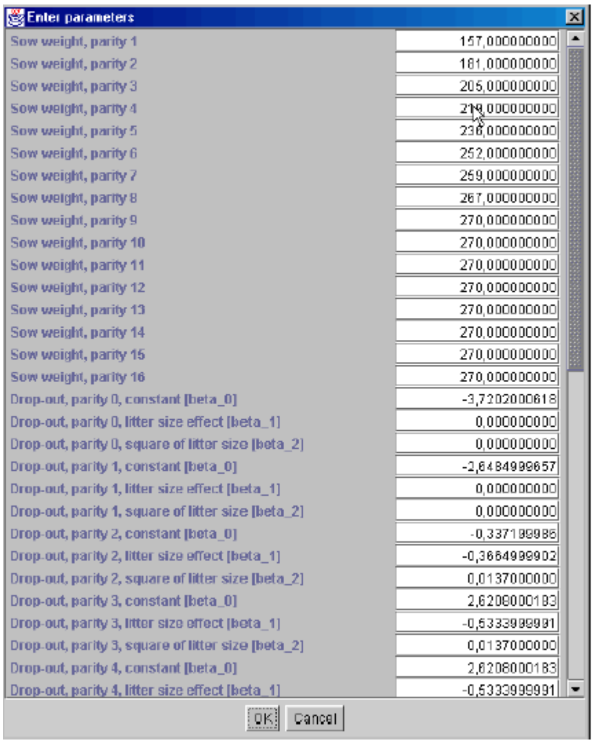
4. ábra. A koca modell paramétereinek beállítása

A 4. ábra mutatja, hogy milyen paramétereket kell beállítani az optimalizálás előtt. Elsőként a várható fialások számát kell megadni (number of parities). Majd az egy fialásra jutó termékenyítések minimális és maximális értékét (lowest and highest number of matings) állítjuk be. Meg kell adni a felhasznált kanok (alternative boartypes) számát is és a szoptatás idejét hetekben (weaning age, weeks).



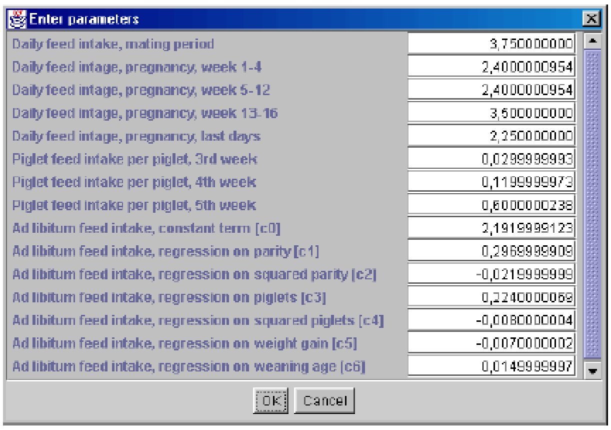
5. ábra. Az alomnagyságát meghatározó modell paramétereinek beállítása

Az 5. ábrán az első négy paraméter a különböző eloszlások átlagait határozza meg, míg a következő kettő a szórásokat definiálja. A 7. sorban a diszkontálási tényezőt lehet beállítani. Az utolsó három sor a technikai környezet és a modell szerkezetének változtatását teszi lehetővé (például itt állíthatjuk be az alomszám alsó és felső határát).



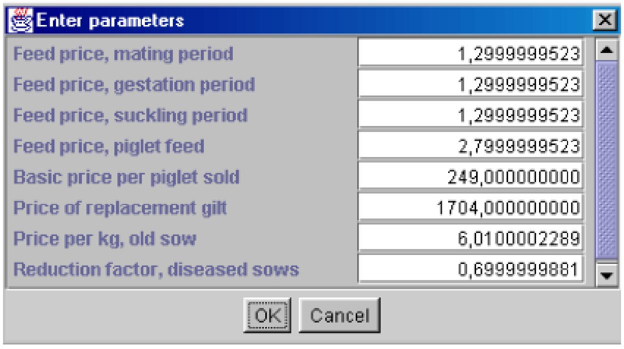
6. ábra. A koca paramétereinek beállítása

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



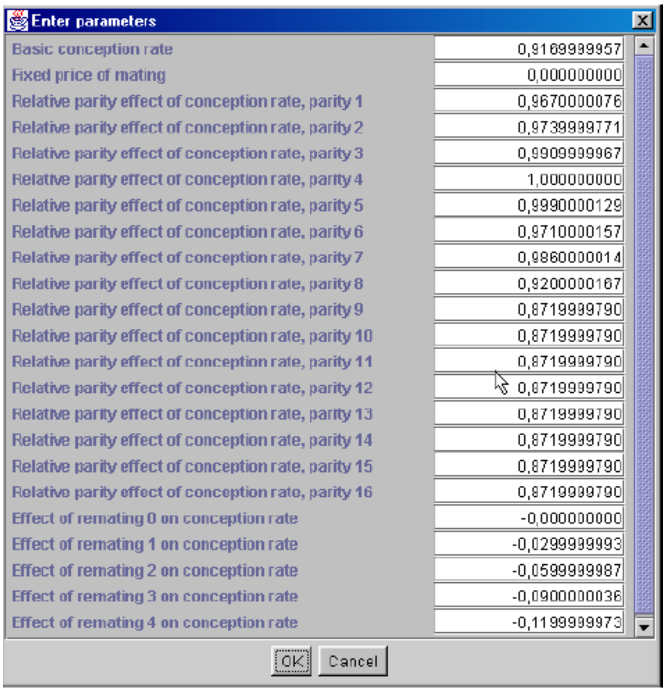
7. ábra. A takarmány fogyasztás paramétereinek beállítása

A 7. ábrán a vemhesség és a szoptatás alatti takarmányfogyasztás (Daily feed intake) 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 (Ad libitum feed intake) regressziós paramétereit is itt tudjuk definiálni $c_0 – c_6$-ig (a fialások számát (parity), a fialások számának négyzetét, a malacszámot (piglets), a malacok számának négyzetét, a tömeggyarapodást (weight gain), a választási kort (weaning age)).



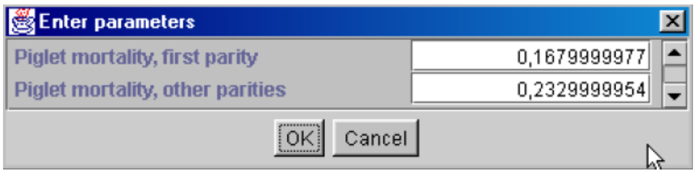
8. ábra. A különböző árak rögzítése

A 8. ábra az árak paramétereinek beállítását mutatja. A koca takarmányának árai (feed price) megadhatók a különböző időszakokra (termékenyítés, vemhesség, szoptatás) és a malactáp (piglet feed) ára is itt rögzíthető. Beállítható a malacár (basic price per piglet sold), a tenyészsüldő ára (replacement gilt), a selejtkoca ár (old sow) és a betegkoca (diseased sow) csökkentett értéke.



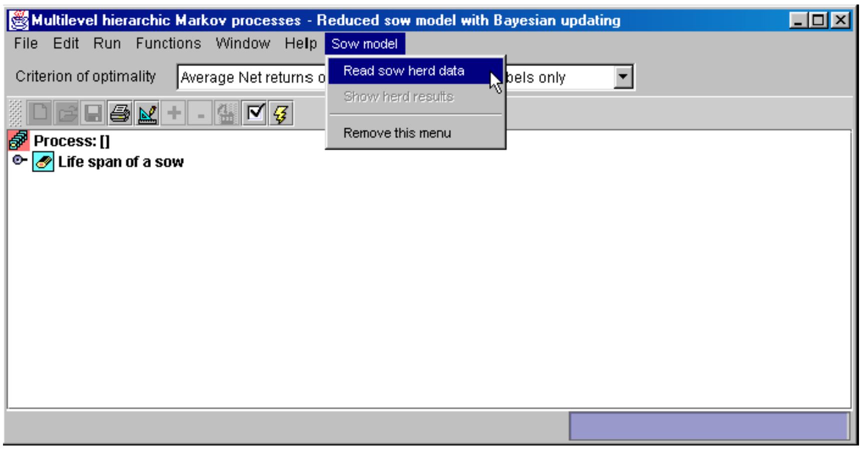
9. ábra. A termékenyítés paramétereinek beállítása

A 9. ábrán a termékenyítés paramétereit mutatjuk be. Minden fialásnál külön – külön megadhatók az értékek. A termékenyítési arányt (conception rate) az alaparánytól (basic conception rate) és a fialástól függő relatívhatásként (relative parity effect) 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 (remating).



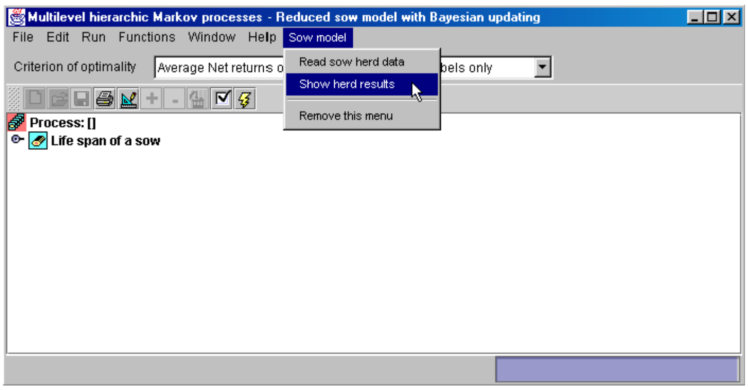
10. ábra. A malacok elhullásának alakulása az első és a későbbi fialások során

A 10. ábrán bemutatjuk a malacok elhullásának (piglet mortality) alakulását az első és a későbbi fialások során.



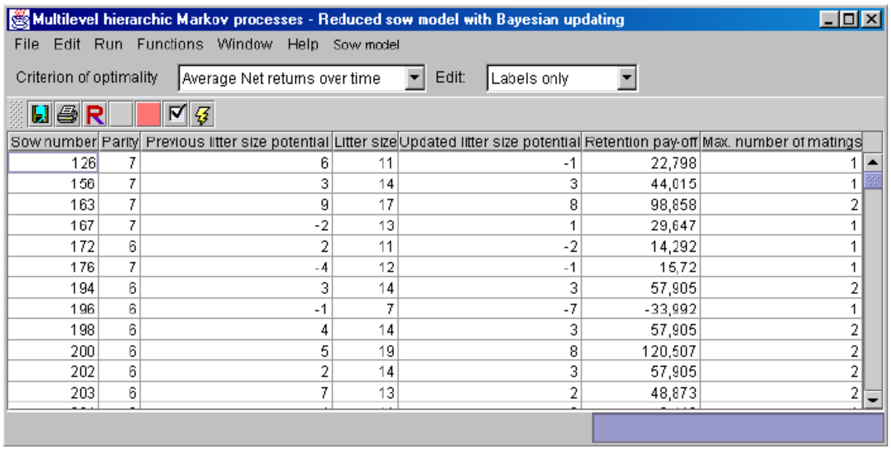
11. ábra. A „Koca modell” főmenüje.

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 (Read sow herd data) megnyissuk (12. ábra). Így lehetővé válik a már megoldott eredmények szakmai vizsgálata.



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	6	2	11	-2	14,292	1
176	7	-4	12	-1	15,72	1
194	6	3	14	3	57,905	2
196	6	-1	7	-7	-33,992	1
198	6	4	14	3	57,905	2
200	6	5	19	8	120,507	2
202	6	2	14	3	57,905	2
203	6	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 (Average Net returns over time) és a kocánkénti döntések sorozatát. Minden egyes kocának meg van az azonosítója (sow number), fialásainak a száma (parity), a lehetséges alomszáma (previous litter size potential), az utolsó alom mérete (litter size). Látható továbbá a potenciális alomszám értéke úgy, hogy figyelembe vettük a legutolsó alom nagyságát (updated litter size potential), a koca termelésben tartásának jövőbeli jövedelmezősége (retention pay-off), a vemhesítések számának maximuma (maximum number of matings) mielőtt a kocát termékenységi zavarok miatt selejtezni kellene. A legérdekesebb információ az elérhető jövedelem (retention pay-off), amelyet tekinthetünk gazdasági indexnek is, 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.

Hivatkozások

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 programing 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 Bayessian 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 Bayessian 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 Rewiew of Agricultural Economics 18: 231-244.

Kristensen, A.R. and Jorgensen, E. 1995. Applicational perspectives of recent developements in dynamic programming methods for herd management support. Dina Notat No. 33.

Ross, S. M. 1970. Applied probability models with optimalization 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.