Kristensen, A. R. 1993. *Markov Decision Programming Techniques Applied to the Animal Replacement Problem*. Dissertation for the dr. agro. degree. Dina KVL, Department of Animal Science and Animal Health, The Royal Veterinary and Agricultural University, Copenhagen. ISBN 87 7026 332 9.

Chapter II. A survey of Markov decision programming techniques applied to the animal replacement problem (p 13-30)

A shorter version of the chapter has been published as

Kristensen, A. R. 1994. A survey of Markov decision programming techniques applied to the animal replacement problem. *European Review of Agricultural Economics 21*, 73-93. DOI: 10.1093/erae/21.1.73

II

A survey of Markov decision programming techniques applied to the animal replacement problem[1]

ANDERS RINGGAARD KRISTENSEN
Department of Animal Science and Animal Health The Royal Veterinary and Agricultural University, Rolighedsvej 23, DK-1958 Frederiksberg C, Copenhagen, Denmark

Abstract
The major difficulties of the animal replacement problem are identified as uniformity, herd restraints and the "curse of dimensionality". Approaches for circumventing these difficulties using Markov decision programming methods are systematically discussed, and possible optimization techniques are described and evaluated. Assuming that the objective of the farmer is maximum net returns from the entire herd, relevant criteria of optimality are discussed. It is concluded that a Bayesian technique is a promising approach as concerns the uniformity problem, that parameter iteration may be used under herd restraints, and that hierarchic Markov processes has contributed to the solution of the dimensionality problem.

Keywords: Criteria of optimality, hierarchic Markov process, parameter iteration, Bayesian updating.

1. Introduction

This paper deals with a *problem* and a *technique*. The problem is the determination of optimal *replacement of animals* (in practice limited to cows and sows). The technique is *dynamic programming* or, to be more specific, *Markov decision programming*. The literature on the replacement problem in *general* is very extensive. Studies on the *animal* replacement problem are also numerous, but naturally they are fewer than for the general problem. A review of studies on dairy cow replacement is given by van Arendonk (1984). Also on Markov decision programming the literature is extensive. Recent reviews are given by van der Wal and Wessels (1985) as well as White and White (1989). A review of applications to agriculture has been given by Kennedy (1981).

Since *both* the problem and the technique discussed in this paper seem to be well elucidated in the literature, a relevant question to ask would be why the *combination* of the problem and the technique should be the subject of a survey. The answer is that animal replacement problems differ from general replacement problems in several respects, and in order to deal with the problems arising from this observation many modifications of the general Markov decision programming technique are relevant or even necessary.

The general replacement theory most often implicitly assumes industrial items as the objects of replacement. Ben-Ari et al. (1983) mention three main features in which the dairy cow replacement problem differs from the industrial problem. Exactly the same features are relevant in sow replacement models.

- *Uniformity*. It is a problem that the traits of an animal are difficult to define and measure. Furthermore the variance of each trait is relatively large.
- *Reproductive cycle*. The production of an animal is cyclic. We therefore need to decide in *which* cycle to replace as well as when to replace *inside* a cycle.
- *Availability*. Usually there is a limited supply of replacements (heifers or gilts). This is particularly the case when the farmer only uses home-

[1] This research was carried out as part of Dina, Danish Informatics Network in the Agricultural Sciences

grown animals – for instance because of infection risks when animals are bought at the market.

The problem of availability is only one example of a restraint that applies to the herd as a whole. Other examples might be a milk quota, a limited supply of roughages or limiting housing capacity. In all cases the animals considered for replacement compete for the resource (or quota) in question. We shall therefore in this study consider the more general problem of optimal replacement under some *herd restraint*.

The main reason for using Markov decision programming in the determination of optimal animal replacement policies is probably the variation in traits, which with this technique is taken into account directly. Also the cyclic production may be directly considered by traditional Markov decision programming. Very soon, however, a problem of *dimensionality* is faced. If *several* traits of the animal are considered simultaneously, and each trait is considered at a realistic number of levels, the *state space* becomes very large (the size of the state space is in principle calculated as the number of traits times the number of levels of each). Even though the method in theory can handle the problem, optimization is prohibitive even on modern computers. In literature, the problem is referred to as the *"curse of dimensionality"*.

The objective of this study is to discuss how the technique (Markov decision programming) may be adapted to solve the problem (the animal replacement problem), where *uniformity* and *herd restraints* as well as the *curse of dimensionality* (arising from the variability in traits and the cyclic production) have been identified as major difficulties to be taken into account. During the decade since the reviews of Kennedy (1981) and van Arendonk (1984) were written, many results have been achieved concerning these difficulties.

We shall assume throughout the study that the objective of the farmer is the maximization of net revenue from the *entire* herd. In each situation, we shall consider how this objective may be transformed to a relevant criterion of optimality to be used in the Markov decision process.

2. Variability and cyclic production: Markov decision programming

As mentioned in the introduction, Markov decision programming is directly able to take the variability in traits and the cyclic production into account without any adaptations. In order to have a frame of reference, we shall briefly present the theory of traditional Markov decision programming originally described by Howard (1960).

2.1. Notation and terminology

Consider a discrete time Markov decision process with a finite *state* space $\mathbf{U} = \{1,2,...,u\}$ and a finite *action* set $\mathbf{D}$. A *policy* s is a map assigning to each state i an action $s(i) \in \mathbf{D}$. Let $p_{ij}^{\ d}$ be the *transition probability* from state i to state j if the action $d \in \mathbf{D}$ is taken. The *reward* to be gained when the state i is observed, and the action d is taken, is denoted as r_i^d. The time interval between two transitions is called a *stage*.

We have now defined the elements of a traditional Markov decision process, but in some models we further assume that if state i is observed, and action d is taken, a physical quantity of m_i^d is involved (e.g. Kristensen, 1989; 1991). In this study we shall refer to m_i^d as the *physical output*. If $s(i) = d$, the symbols r_i^d, m_i^d and $p_{ij}^{\ d}$ are also written as r_i^s, m_i^s and $p_{ij}^{\ s}$, respectively.

An *optimal* policy is defined as a policy that maximizes (or minimizes) some predefined objective function. The optimization technique (i.e. the method to identify an optimal policy) depends on the form of the objective function or – in other words – on the criterion of optimality. The over-all objective to maximize net revenue of the entire herd may (depending on the circumstances) result in different criteria of optimality formulated as alternative objective functions. The choice of criterion depends on whether the planning horizon is finite or infinite.

2.2. Criteria of optimality

2.2.1. Finite planning horizon

A farmer, who knows that he is going to terminate his production after N stages, may use the maximization of total expected rewards as his criterion

of optimality. The corresponding objective function h is

$$h(s^1,\ldots,s^N) = \mathrm{E}\left(\sum_{n=1}^{N} r_{I(n)}^{s^n}\right), \quad (1)$$

where E denotes the expected value, s^n is the policy at stage n, and $I(n)$ is the (unknown) state at stage n.

If the farmer has a time preference, so that he prefers an immediate reward to an identical reward later on, a better criterion is the maximization of total expected discounted rewards. If all stages are of equal length, this is equal to applying the objective function

$$h(s^1,\ldots,s^N) = \mathrm{E}\left(\sum_{n=1}^{N} \beta^{n-1} r_{I(n)}^{s^n}\right), \quad (2)$$

where $\beta < 1$ is the discount factor defined by the interest rate and the stage length.

2.2.2. *Infinite planning horizon*

A situation where the stage of termination is unknown (but at least far ahead) is usually modeled by an infinite planning horizon (i.e. $N = \infty$). In that case the optimal policy is constant over stages. The function (1) cannot be applied in this situation, but since $\beta < 1$, the function (2) will converge towards a fixed value for N becoming very large. Thus the objective function is given by

$$h(s) = \mathrm{E}\left(\sum_{n=1}^{\infty} \beta^{n-1} r_{I(n)}^{s}\right). \quad (3)$$

Since, usually, each animal and its future successors are represented by a separate Markov decision process, this criterion together with the criterion (2), are equal to the maximization of total discounted net revenues *per animal*. Such a criterion is relevant in a situation where a limiting housing capacity is the only (or at least the *most* limiting) herd restraint.

An alternative criterion under infinite planning horizon is the maximization of expected average reward per unit of time. If all stages are of equal length, the objective function in this situation is

$$h(s) = g^s = \sum_{i=1}^{u} \pi_i^s r_i^s, \quad (4)$$

where π_i^s is the limiting state probability under the policy s (i.e. when the policy is kept constant over an infinite number of stages). This criterion maximizes the average net revenues per stage, i.e. *over time*. It may be relevant under the same conditions as criterion (3) if an animal and its future successors are represented by a separate Markov decision process. Practical experience shows that the optimal replacement policies determined under criteria (3) and (4) are almost identical.

If a herd restraint (e.g. a milk quota) is imposed on the physical output, a relevant criterion may be the maximization of expected average reward per unit of physical output using the objective function

$$h(s) = g^s = \frac{g_r^s}{g_m^s} = \frac{\sum_{i=1}^{u} \pi_i^s r_i^s}{\sum_{i=1}^{u} \pi_i^s m_i^s}. \quad (5)$$

In case of a milk quota, the physical output m_i^s is the milk produced by a cow in state i under policy s. The function (5) is also relevant if the criterion is the maximization of the expected average reward over time in a model where the stage length varies. In that case the physical output represents the stage length. It should be noticed that if $m_i^d = 1$ for all i and d, the function (5) is identical to (4). The symbol g_r^s is the average reward over stages (equal to g^s of Eq. (4)) and g_m^s is the average physical output over stages.

2.3. *Optimization techniques in general Markov decision programming*

2.3.1. *Value iteration*

Under finite planning horizon the *value iteration method* is excellent. The optimal policies are determined sequentially using the functional equations

$$f_i(n) = \max_d \left\{ r_i^d + \beta \sum_{j=1}^{u} p_{ij}^d f_j(n-1) \right\}, \quad i=1,\ldots,u, \quad (6)$$

where the action d maximizing the right hand side is optimal for state i at the stage in question. The function $f_i(n)$ is the total expected discounted rewards from the process when it starts from state i and will operate for n stages before termination. Thus $f_i(0)$ is the salvage value of the system when it is in state i. At each stage an optimal policy is

chosen using Eqs. (6). If the objective function (1) is used, $\beta = 1$ in Eq. (6). Otherwise β is the discount factor.

Under infinite planning horizon, the value iteration method may be used to approximate an optimal policy. Under the objective function (3) it is possible to show that (cf. Howard 1960)

$$\lim_{n\to\infty} f_i(n) = f_i\ ,\quad i=1,...,u\ , \tag{7}$$

where f_i for fixed i is a constant. By using Eqs. (6) over a large number of stages, we will sooner or later observe that $f_i(n+1)$ is almost equal to $f_i(n)$ for all i. Further we will observe that the same policy is chosen during several stages. We can feel rather sure that such a policy is close to be optimal, but there is no guarantee that it is identical to an optimal policy. For practical purposes, however, the approximation usually suffices.

Since the objective function (4) is just a special case of function (5), where $m_i^s = 1$ for all i and d, we shall only consider the criterion given by (5). In this case $f_i(n)$ is the total expected rewards when the process starts from the beginning of a stage in state i and will operate *until n units of physical output have been produced.* Under the criterion given by the objective function (4), the production of n units of output is just the operation of the process over n stages. It is assumed that the physical output only takes integer values (for practical purpose this is just a question of selecting an appropriate unit). According to Howard (1971) an optimal policy for producing n units of output (i.e. a policy that maximizes the expected reward of producing n units) is determined recursively by the relations ($i=1,...,u$):

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

where

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

This is under the assumption that the reward/output rate has the constant value of r_i^d/m_i^d during the entire stage. However, since the physical output is bounded, it is easily seen that for n sufficiently large, $a = 0$. Hence we get for $i=1,...,u$

$$f_i(n) = \max_d \left\{ r_i^d + \sum_{j=1}^{u} p_{ij}^d f_j(n-m_i^d)\right\},\quad large\ n\ . \tag{8}$$

Thus in the long run, the assumption concerning constant reward/output rate in all states will have no effect. The equivalence of Eq. (7) is in this case

$$\lim_{n\to\infty}(f_i(n) - f_i(n-1)) = g\ , \tag{9}$$

and sooner or later the policy will not differ from step to step of Eqs. (8).

Further details on the value iteration method are given by Howard (1960; 1971). It should particularly be noticed that m_i^d, which in this study is interpreted as a physical output (e.g. milk yield), in the study of Howard (1971) is interpreted as the expected *stage length* when state i is observed under the action d. Thus in his model the criterion (5) is the expected average reward over time. Compared to Eq. (8), Howard (1971) described a more general case where the stage length is a random variable of which the distribution is given by the action and the present state as well as the state to be observed at the next stage. Further the reward depends on the state combination, the action and the stage length. The interpretation as physical output has been discussed by Kristensen (1991).

The value iteration method is identical to what is usually referred to as *dynamic programming*, *successive iteration* or *successive approximation.*

2.3.2. Policy iteration

Under infinite planning horizon, the *policy iteration method* may be applied. Unlike the value iteration method it always provides an optimal policy. It covers all three objective functions (3), (4) and (5). The iteration cycle used for optimization has the following steps:

1) Choose an arbitrary policy s. Go to 2.
2) Solve the set of linear simultaneous equations appearing in Table 1. Go to 3.
3) For each state i, find the action d' that maximizes the expression given in Table 1, and put $s'(i)=d'$. If $s'=s$ then stop, since an optimal policy is found. Otherwise redefine s according to the new policy (i.e. put $s=s'$) and go back to 2.

Table 1. Equations and expressions to be used in the policy iteration cycle with different objective functions.

Objective function	Linear equations of Step 2			Expression Step 3
	Equation (i=1,...,u)	Unknowns	Additional equation	
(3)	$f_i^s = r_i^s + \beta \sum_{j=1}^{u} p_{ij}^s f_j^s$	$f_1^s,\ldots,f_u^s$	–	$r_i^d + \beta \sum_{j=1}^{u} p_{ij}^d f_j^s$
(4)	$g^s + f_i^s = r_i^s + \sum_{j=1}^{u} p_{ij}^s f_j^s$	$g^s, f_1^s,\ldots,f_u^s$	$f_u^s = 0$	$r_i^d + \sum_{j=1}^{u} p_{ij}^d f_j^s$
(5)	$g^s m_i^s + f_i^s = r_i^s + \sum_{j=1}^{u} p_{ij}^s f_j^s$	$g^s, f_1^s,\ldots,f_u^s$	$f_u^s = 0$	$r_i^d - m_i^d g^s + \sum_{j=1}^{n} p_{ij}^d f_j^s$

From the equations and expressions of Table 1, we see that also with the policy iteration method the objective function (4) is just a special case of (5) where $m_i^s = 1$ for all i and d. For the objective functions (3) and (4) the policy iteration method was developed by Howard (1960), and for the function (5) a policy iteration method was presented by Jewell (1963). Like Howard (1971), Jewell interpreted m_i^d as the expected stage length.

Under Criterion (3), f_i^s is the total present value of the expected future rewards of a process starting in state i and running over an infinite number of stages following the constant policy s. Under Criterions (4) and (5), f_i^s is the *relative value* of state i under the policy s. The difference in relative values between two states equals the amount of money a rational person is just willing to pay in order to start in the highest ranking of the two states instead of the lowest ranking. The absolute value of f_i^s is determined arbitrarily by the additional equation of Table 1, where the relative value of state u is defined to be zero. The interpretation of relative values is discussed in details by Kristensen (1991).

2.3.3. Linear programming

Under an infinite planning horizon, linear programming is a possible optimization technique. When the criterion (3) is applied the linear programming problem becomes (cf. Ross, 1970)

$$\sum_{i=1}^{u} x_i = Max!$$

subject to (10)

$$x_i - \beta \sum_{j=1}^{u} p_{ij}^d x_j \geq r_i^d, \quad all\ d \in D,\ i=1,\ldots,u.$$

It appears from (10) that each combination of state and action is represented by exactly one restriction. An action d is optimal in state i if, and only if, the corresponding restriction is satisfied as an equation when the values of $x_1,\ldots,x_u$ arises from an optimal solution to the linear programming problem. The *optimal* values of $x_1,\ldots,x_u$ are equal to the present values $f_1^s,\ldots,f_u^s$ under an optimal policy.

If the objective function (4) is applied, the linear programming problem becomes

$$\sum_{i=1}^{u} \sum_{d \in D} r_i^d x_i^d = Max!$$

subject to

$$\sum_{d \in D} x_i^d - \sum_{j=1}^{u} \sum_{d \in D} p_{ij}^d x_j^d = 0, \quad i=1,\ldots,u \qquad (11)$$

$$\sum_{i=1}^{u} \sum_{d \in D} x_i^d = 1$$

$$x_i^d \geq 0, \quad d \in D,\ i=1,\ldots,u.$$

In this case an action d is optimal in state i if and only if x_i^d from the optimal solution is strictly positive. The optimal value of the objective function is equal to the average rewards per stage under an optimal policy. The optimal value of $\Sigma_{d \in D} x_i^d$ is

equal to the limiting state probability π_i under an optimal policy.

Using Criterion (5), we may solve the following linear programming problem (cf. Kennedy,1986):

$$x_u = Max!$$

subject to

$$-x_i + \sum_{j=1}^{u-1} p_{ij}^d x_j - m_i^d x_u \leq -r_i^d - p_{iu}^d a \ , \qquad (12)$$

$$d \in D \ , \ i = 1,\ldots,u-1$$

$$\sum_{j=1}^{u-1} p_{uj}^d x_j - m_u^d x_u \leq -r_u^d - p_{uu}^d a + a \ , \quad d \in D$$

$$x_i \geq 0 \ , \ i = 1,\ldots,u \ ,$$

where a is a pre-determined relative value of state u chosen sufficiently large to ensure that all other relative values are positive. The optimal value of the objective function of the linear programming problem is equal to the expected average reward per unit of output as defined in Eq. (5) under an optimal policy. The optimal values of the variables $x_1,\ldots,x_{u-1}$ are equal to the relative values of the states 1,...,u-1, provided that the relative value of state u is equal to a . As it appears, each combination of state and action is represented by one and only one restriction. An action is optimal in a state if and only if the corresponding restriction is satisfied as an equation in the optimal solution.

Since Criterion (4) is just a special case of (5) with all physical outputs set to the value 1, the linear programming problem (12) may also be used in the determination of an optimal policy under Criterion (4).

2.4. Discussion and applications

Under finite planning horizon, the value iteration method is perfect, but in replacement models the planning horizon is rarely well defined. Most often the process is assumed to operate over an unknown period of time with no pre-determined stage of termination. In such cases the abstraction of an infinite planning horizon seems more relevant. Therefore we shall pay specific attention to the optimization problem under the criteria (3), (4) and (5) where all three techniques described in the previous sections are available.

The value iteration method is not exact, and the convergence is rather slow. On the other hand, the mathematical formulation is very simple, and the method makes it possible to handle very large models with thousands of states. Further it is possible to let the reward and/or the physical output depend on the stage number in some pre-defined way. This has been mentioned by van Arendonk (1984) as an advantage in modelling genetic improvement over time. The method has been used in a lot of dairy cow replacement models as an approximation to the infinite stage optimum. Thus it has been used by Jenkins and Halter (1963), Giaever (1966), Smith (1971), McArthur (1973), Steward et al. (1977; 1978), Killen and Kearney (1978), Ben-Ari et al. (1983), van Arendonk (1985; 1986) and van Arendonk and Dijkhuizen (1985). Some of the models mentioned have been very large. For instance, the model of van Arendonk and Dijkhuizen contained 174 000 states (reported by van Arendonk, 1988). In sows, the method has been used by Huirne et al. (1988).

The policy iteration method has almost exactly the opposite characteristics of the value iteration method. Because of the more complicated mathematical formulation involving solution of large systems of simultaneous linear equations, the method can only handle rather small models with, say, a few hundred states. The solution of the linear equations implies the inversion of a matrix of the dimension $u \times u$, which is rather complicated. On the other hand, the method is exact and very efficient in the sense of fast convergence. The rewards are not allowed to depend on the stage except for a fixed rate of annual increase (e.g. inflation) or decrease. However, a seasonal variation in rewards or physical outputs is easily modeled by including a state variable describing season (each state is usually defined by the value of a number of state variables describing the system).

An advantage of the policy iteration method is that the equations in Table 1 are *general*. Under any policy s we are able to calculate directly the economic consequences of following the policy by solution of the equations. This makes it possible to compare the economic consequences of various non-optimal policies to those of the optimal. Further we may use the equations belonging to the criterion (5) to calculate the long run technical results under a given policy by redefining r_i^s and m_i^s . If for instance $r_i^s = 1$ if a calving takes place and zero otherwise, and m_i^s is the stage length when

state i is observed under policy s, then g^s, which is the average number of calvings per cow per year, may be determined from the equations. Further examples are discussed by Kristensen (1991). For an example where the equations are used for calculation of the economic value of culling information, reference is made to Kristensen and Thysen (1991).

The policy iteration method has been used by Reenberg (1979) and Kristensen and Østergaard (1982). The models were very small, containing only 9 and 177 states, respectively.

3. The curse of dimensionality: Hierarchic Markov processes

In order to combine the computational advantages of the value iteration method with the exactness and efficiency of the policy iteration method Kristensen (1988; 1991) introduced a new notion of a hierarchic Markov process. It is a contribution to the solution of the problem referred to as the "curse of dimensionality" since it makes it possible to give exact solutions to models with even very large state spaces. A hierarchic Markov process is only relevant under infinite planning horizon, and there is no relevance of the criterion (4) because the special situation where the physical output equals 1 in all stages has no computational advantages over other values. Therefore we shall only consider the criteria (3) and (5).

3.1. Notation and terminology

A hierarchic Markov process is a series of Markov decision processes called *subprocesses* built together in one Markov decision process called the *main process*. A subprocess is a finite time Markov decision process with N stages and a finite state space $\Omega_n = \{1,...,u_n\}$ for stage n, $1 \le n \le N$. The action set $\mathbf{D}_n$ of the nth stage is assumed to be finite, too. A policy s of a subprocess is a map assigning to each stage n and state $i \in \Omega_n$ an action $s(n,i) \in \mathbf{D}_n$. The set of all possible policies of a subprocess is denoted Γ. When the state i is observed and the action d is taken, a reward $r_i^d(n)$ is gained. The corresponding physical output is denoted as $m_i^d(n)$. Let $p_{ij}^s(n)$ be the transition probability from state i to state j where i is the state at the nth stage, j is the state at the following stage and d is the action taken at stage n. Under the Criterion (3) we shall denote the discount factor in state i under the action d as $\beta_i^d(n)$ assuming that the stage length is given by stage, state and action.

Assume that we have a set of ν possible subprocesses each having its own individual set of parameters. The main process is then a Markov decision process running over an infinite number of stages and having the finite state space $\{1,...,\nu\}$. Each stage in this process represents a particular subprocess. The action sets of the main process are the sets Γ_ι, $\iota = 1,...,\nu$, of all possible policies of the individual subprocesses (to avoid ambiguity the states of the main process will be denoted by Greek letters ι, κ etc.). A policy σ is a map assigning to each state ι of the main process an action $\sigma(\iota) \in \Gamma_\iota$. The transition matrix of the main process has the dimension $\nu \times \nu$, and it is denoted $\Phi = \{\phi_{\iota\kappa}\}$. The transition probabilities are assumed to be independent of the action taken. The reward f_ι^σ and the physical output h_ι^σ in state ι of the main process are determined from the total rewards and output functions of the corresponding subprocess

$$f_i^s(n) = r_i^s(n), \quad n=N$$

$$f_i^s(n) = r_i^s(n) + \beta_i^s(n) \sum_{j=1}^{u_{n+1}} p_{ij}^s(n) f_j^s(n+1), \quad n=1,...,N-1, \tag{13}$$

and

$$f_\iota^\sigma = \sum_{i=1}^{u_1} p_i(0) f_i^s(1), \quad s=\sigma(\iota),$$

and analogously for h_ι^σ (except for the discount factor). The symbol $p_i(0)$ is the probability of observing state i at the first stage of the subprocess. Finally, the expected discount factor in state ι under the action s is denoted as B_ι^s and calculated as follows

$$b_i^s(n) = \beta_i^s(n), \quad n=N$$

$$b_i^s(n) = \beta_i^s(n) \sum_{j=1}^{u_{n+1}} p_{ij}^s(n) b_j^s(n+1), \quad n=1,...,N-1, \tag{14}$$

and

$$B_\iota^s = \sum_{i=1}^{u_1} p_i(0) b_i^s(1).$$

3.2. Optimization

Since the main process is just an ordinary Markov decision process, the policy iteration cycle described in Section 2.3.2 might be used directly for optimization. In practice Steps 1 and 2 are easily carried out, but Step 3 is prohibitive because of the extremely large number of alternative actions $s \in \Gamma_\iota$ (as mentioned above s is an entire policy of the ιth subprocess). To circumvent this problem Kristensen (1988; 1991) constructed an iterative method, where a value iteration method is applied in the subprocesses and the results are used in Step 3 of the policy iteration method of the main process. The different versions of the method covers the criteria of optimality under infinite planning horizon defined as (3) and (5) in Section 2.2.2. Since criterion (4) is a special case of (5) it is also indirectly covered.

The general form of the iteration cycle of a hierarchic Markov process has the following steps:

1) Choose an arbitrary policy σ. Go to 2.
2) Solve the following set of linear simultaneous equations for $F_1^\sigma,...,F_\nu^\sigma$ and in case of Criterion (5) for g^σ:

$$g^\sigma h_\iota^\sigma + F_\iota^\sigma = f_\iota^\sigma + B_\iota^\sigma \sum_{\kappa=1}^{\nu} \phi_{\iota\kappa} F_\kappa^\sigma , \quad \iota = 1,...,\nu .$$

   In case of Criterion (5) the additional equation $F_\nu^\sigma = 0$ is necessary in order to determine a unique solution. Go to 3.
3) Define

$$T_\iota = \sum_{\kappa=1}^{\nu} \phi_{\iota\kappa} F_\kappa^\sigma$$

   under Criterion (3) and $T_\iota = 0$ under Criterion (5). For each subprocess ι, find by means of the recurrence equations

$$\tau_{\iota,i}(n) = \max_d \left\{ r_i^d(n) - m_i^d(n) g^\sigma + \beta_i^d(N) T_\iota \right\} ,$$
$$n = N$$

$$\tau_{\iota,i}(n) = \max_d \left\{ r_i^d(n) - m_i^d(n) g^\sigma + \beta_i^d(n) \sum_{j=1}^{u_{n+1}} p_{ij}^d(n) \tau_{\iota,j}(n+1) \right\} , \quad n = 1,...,N-1 .$$

   a policy s' of the subprocess. The action $s'(n,i)$ is equal to the d' that maximizes the right hand side of the recurrence equation of state i at stage n. Put $\sigma'(\iota) = s'$ for $\iota = 1,...,\nu$. If $\sigma' = \sigma$, then stop since an optimal policy is found. Otherwise, redefine σ according to the new policy (i.e. put $\sigma = \sigma'$) and go back to 2.

When the iteration cycle is used under Criterion (3) all physical outputs ($m_i^d(n)$ and accordingly also h_ι^σ) are put equal to zero. The iteration cycle covering this situation was developed by Kristensen (1988).

Under Criterion (4) all physical outputs $m_i^d(n)$ and all discount factors $\beta_i^d(n)$ and B_ι^σ are put equal to 1, but under Criterion (5) only the discount factors are put equal to 1. The iteration cycle covering these situations was described by Kristensen (1991).

3.3. Discussion and applications

The hierarchic Markov process is specially designed to fit the structure of replacement problems where the successive stages of the subprocesses correspond to the age of the asset in question. By appropriate selection of state spaces in the subprocesses and the main process it is possible to find optimal solutions to even very large models. The idea is to let the number of states in the subprocesses (where a value iteration technique is applied) be very large and only include very few states in the main process (where the technique is directly based on the policy iteration method). Thus we have got a method which is at the same time fast, exact and able to handle very large models.

Kristensen (1987) used the technique in a dairy cow replacement model which in a traditional formulation as an ordinary Markov decision process would have contained approximately 60 000 states, and later (Kristensen, 1989) in a model with approximately 180 000 states. In both cases the number of states in the main process was only 5, reducing Step 2 to the solution of only 5 simultaneous linear equations (versus 180 000 in a traditional formulation). Even in these very large models the number of iterations needed to provide an optimal solution was only from 3 to 6 (tested under 100 different price and production conditions, Kristensen, 1991). Recently, the method is applied by Houben et al. (1992).

In sows, Huirne et al. (1992) seem to have applied a technique which in many aspects is similar to a hierarchic Markov process, but they have not explained their method in all details. Also Jør-

gensen (1992a) has applied a technique which is inspired of a hierarchic Markov process in a sow replacement model, and recently (Jørgensen 1992b), he used the hierarchic method in the determination of optimal delivery policies in slaughter pigs.

Naturally the hierarchic model just described may also be formulated as an ordinary Markov decision process. In that case each combination of subprocess (main state), stage and state should be interpreted as a state. We shall denote a state in the transformed process as (ιni), and the parameters are

$$\begin{aligned} r^d_{\iota ni} &= r^d_i(n)\,, \\ m^d_{\iota ni} &= m^d_i(n)\,, \\ \beta^d_{\iota ni} &= \beta^d_i(n)\,, \end{aligned} \tag{15}$$

$$p^d_{(\iota ni)(\kappa mj)} = \begin{cases} p^d_{ij}(n), & \iota=\kappa \ \wedge\ m=n-1 \\ \phi_{\iota\kappa}\, p_i(0), & n=N \wedge m=1 \\ 0, & \textit{otherwise} \end{cases},$$

where the parameters mentioned on the right hand side of the equations are those belonging to the ιth subprocess except for $p_i(0)$ which belongs to subprocess κ . This formulation of course has the same optimal policies as the hierarchic formulation, so it is only computational advantages that make the hierarchic model relevant. A comparison to traditional methods may therefore be relevant.

Since the policy iteration method involves the solution of a set of u equations (where u is the number of states) it is only relevant for small models. The value iteration method, however, has been used with even very large models and may handle problems of the same size as the hierarchic formulation, but the time spent on optimization is much lower under the hierarchic formulation. To recognize this, we shall compare the calculations involved.

Step 3 of the hierarchic optimization involves exactly the same number of operations as one iteration of the value iteration method (Eq. (6)). The further needs of the hierarchic method are the calculation of the rewards and **either** the physical output **or** the expected discount factor of a stage in the main process according to Eqs. (13) and (14). Since the calculations at each stage is only carried out for one action, the calculation of both main state parameters involves approximately the same number of operations as one iteration under the value iteration method if the number of alternative actions is 2. If the number of actions is higher, the calculations relatively involves a lower number of operations than an iteration under the value iteration method. These considerations are based on the assumption that the value iteration method is programmed in an efficient way, so that the sum of Eq. (6) is not calculated as a sum of all u elements, but only as a sum of those elements where $p_{ij}{}^d$ is not zero according to Eq. (15). Otherwise the hierarchic technique will be even more superior. Finally the system of linear equations of Step 2 of the hierarchic cycle must be solved, but in large models with only a few states in the main process the time spent on this is negligible.

If we use the considerations above in a practical example, the advantages of the hierarchic technique becomes obvious. As reported by Kristensen (1991) a model with 180 000 state combinations was optimized by the hierarchic technique under 100 different price conditions. The number of iterations needed ranged from 3 to 6 corresponding to between 6 and 12 iterations of the value iteration method. If the latter method was used instead, a planning horizon of 20 years would be realistic (cf. van Arendonk 1985). Since each state in the model equals 4 weeks, this horizon represents 260 iterations, which should be compared to the equivalence of from 6 to 12 when the hierarchic technique was applied.

3.4. A numerical example of a hierarchic Markov process

Consider an asset (e.g. a dairy cow) producing two kinds of output items (1 and 2, e.g. milk and beef). We shall assume that the production level of item 1 may change stochastically over time, whereas the production of item 2 is constant over the entire life time of the asset (but may vary between individual assets). At regular time intervals (stages) the asset is inspected in order to determine the production level of item 1. At the first inspection of the asset the production level of item 2 is also determined. In both cases we assume that the result may be "bad", "normal" or "good" (representing the production of 5, 6 and 7 units of item 1 or 3, 4 and 5 units of item 2). After inspection we can choose to keep the asset for at least one additional stage, or

Table 2. Parameters of the hierarchic Markov process, subprocesses.

Sub-pr.	Stg.	St.	$p_{ij}^{1}(n)$				$m_i^1(n)$	$r_i^1(n)$	$p_{ij}^{2}(n)$				$m_i^2(n)$	$r_i^2(n)$
ι	n	i	j=1	j=2	j=3	j=4			j=1	j=2	j=3	j=4		
1	1	1	0.6	0.3	0.1	0.0	5	7	0.0	0.0	0.0	1.0	5	5
1	1	2	0.2	0.6	0.2	0.0	6	8	0.0	0.0	0.0	1.0	6	6
1	1	3	0.1	0.3	0.6	0.0	7	9	0.0	0.0	0.0	1.0	7	7
1	1	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
1	2	1	0.6	0.3	0.1	0.0	5	6	0.0	0.0	0.0	1.0	5	4
1	2	2	0.2	0.6	0.2	0.0	6	7	0.0	0.0	0.0	1.0	6	5
1	2	3	0.1	0.3	0.6	0.0	7	8	0.0	0.0	0.0	1.0	7	6
1	2	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
1	3	1	0.6	0.3	0.1	0.0	5	5	0.0	0.0	0.0	1.0	5	3
1	3	2	0.2	0.6	0.2	0.0	6	6	0.0	0.0	0.0	1.0	6	4
1	3	3	0.1	0.3	0.6	0.0	7	7	0.0	0.0	0.0	1.0	7	5
1	3	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
1	4	1	–	–	–	–	5	2	–	–	–	–	5	2
1	4	2	–	–	–	–	6	3	–	–	–	–	6	3
1	4	3	–	–	–	–	7	4	–	–	–	–	7	4
1	4	4	–	–	–	–	0	0	–	–	–	–	0	0
2	1	1	0.6	0.3	0.1	0.0	5	8	0.0	0.0	0.0	1.0	5	6
2	1	2	0.2	0.6	0.2	0.0	6	9	0.0	0.0	0.0	1.0	6	7
2	1	3	0.1	0.3	0.6	0.0	7	10	0.0	0.0	0.0	1.0	7	8
2	1	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
2	2	1	0.6	0.3	0.1	0.0	5	7	0.0	0.0	0.0	1.0	5	5
2	2	2	0.2	0.6	0.2	0.0	6	8	0.0	0.0	0.0	1.0	6	6
2	2	3	0.1	0.3	0.6	0.0	7	9	0.0	0.0	0.0	1.0	7	7
2	2	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
2	3	1	0.6	0.3	0.1	0.0	5	6	0.0	0.0	0.0	1.0	5	4
2	3	2	0.2	0.6	0.2	0.0	6	7	0.0	0.0	0.0	1.0	6	5
2	3	3	0.1	0.3	0.6	0.0	7	8	0.0	0.0	0.0	1.0	7	6
2	3	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
2	4	1	–	–	–	–	5	3	–	–	–	–	5	3
2	4	2	–	–	–	–	6	4	–	–	–	–	6	4
2	4	3	–	–	–	–	7	5	–	–	–	–	7	5
2	4	4	–	–	–	–	0	0	–	–	–	–	0	0
3	1	1	0.6	0.3	0.1	0.0	5	9	0.0	0.0	0.0	1.0	5	7
3	1	2	0.2	0.6	0.2	0.0	6	10	0.0	0.0	0.0	1.0	6	8
3	1	3	0.2	0.3	0.6	0.0	7	11	0.0	0.0	0.0	1.0	7	9
3	1	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
3	2	1	0.6	0.3	0.1	0.0	5	8	0.0	0.0	0.0	1.0	5	6
3	2	2	0.2	0.6	0.2	0.0	6	9	0.0	0.0	0.0	1.0	6	7
3	2	3	0.1	0.3	0.6	0.0	7	10	0.0	0.0	0.0	1.0	7	8
3	2	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
3	3	1	0.6	0.3	0.1	0.0	5	7	0.0	0.0	0.0	1.0	5	5
3	3	2	0.2	0.6	0.2	0.0	6	8	0.0	0.0	0.0	1.0	6	6
3	3	3	0.1	0.3	0.6	0.0	7	9	0.0	0.0	0.0	1.0	7	7
3	3	4	0.0	0.0	0.0	1.0	0	0	0.0	0.0	0.0	1.0	0	0
3	4	1	–	–	–	–	5	4	–	–	–	–	5	4
3	4	2	–	–	–	–	6	5	–	–	–	–	6	5
3	4	3	–	–	–	–	7	6	–	–	–	–	7	6
3	4	4	–	–	–	–	0	0	–	–	–	–	0	0

we can choose to replace it at the end of the stage at some additional cost.

The three classes of production level of item 2 are defined as states in the main process of a hierarchic Markov process. Thus the number of subprocesses is also 3 and each subprocess represents an asset of a certain productivity concerning item 2. When a new asset is purchased, we assume that the probability distribution over main states is uniform, so that the probability of entering either one is 1/3. The maximum age of an asset is assumed to be 4 stages, and the states of the subprocess are defined from the productivity concerning item 1. Further a dummy state of length, reward and output equal to 0 is included at each stage of the subprocesses. If the asset is replaced at the end of a stage, the process enters the dummy state with probability 1 at the next stage, and for the rest of the duration of the subprocess it will stay in the dummy states.

For all subprocesses we assume that, if the asset is kept, the probability of staying at the same productivity level (state in the subprocess) concerning item 1 is 0.6, and if the present state is "normal", the probability of transition to either "bad" or "good" is 0.2 each. The probability of transition (if kept) from "bad" or "good" to "normal" is in both cases 0.3, and from "bad" to "good" and vice versa the probability is 0.1. The initial state probabilities of the subprocesses are assumed to depend on the subprocess in such a way that for subprocess number 1 (low productivity of item 2) the probabilities of entering state "bad", "normal" and "good" are 0.6, 0.3 and 0.1 respectively. For subprocess number 2 the corresponding probabilities are 0.2, 0.6 and 0.2 and finally for subprocess number 3 they are 0.1, 0.3, 0.6.

The physical output $m_i^d(n)$ of state i at stage n of subprocess number ι is equal to the production of item 1 under the action d, and the corresponding rewards are assumed to be defined as follows:

$$r_{\iota,i}^d(n) = c_1 m_i^d + c_2 k_\iota - c_n - c_3^d, \quad \iota=1,2,3, \quad n=1,\ldots,4, \quad i=1,\ldots,4, \quad d=1,2, \tag{16}$$

where c_1 is the price of item 1, c_2 is the price of item 2, c_n is the cost of operating the asset at the age n, k_ι is the production of item 2 in subprocess (main state) number ι and c_3 is the replacement cost which is zero if no replacement takes place. The cost of operating the asset is assumed to increase linearly from 1 to 4 over stages. Defining $c_1 = c_2 = 1$ and $c_3^2 = 2$ gives us the final parameters appearing in Tables 2 and 3. All stages (except those where the process is in a dummy state of zero length) are assumed to be of equal length, which we for convenience put equal to 1.

We shall determine an optimal solution under the following 3 criteria of optimality:

1) Maximization of total expected discounted rewards, i.e., the objective function (3). In this case the physical outputs of Table 2 are ignored, and a discount factor $\beta_i^d(n) = \exp(-r)$, where r is the interest rate, is applied (for states where the stage length is not zero).

2) Maximization of average rewards over time. In this situation we use the objective function (5) letting the output represent stage length. No discounting is performed in this case.

3) Maximization of average rewards over output defined as in Table 2. Thus the objective function (5) is applied, and no discounting is performed.

Table 3. Parameters of the hierarchic process. Transition probabilities of main process and initial state probabilities of subprocesses

Transition probabilities, main process				Initial state probabilities, subprocesses			
Main state ι	$\phi_{\iota\kappa}$			$p_i(0)$			
	κ=1	κ=2	κ=3	i=1	i=2	i=3	i=4
1	1/3	1/3	1/3	0.6	0.3	0.1	0.0
2	1/3	1/3	1/3	0.2	0.6	0.2	0.0
3	1/3	1/3	1/3	0.1	0.3	0.6	0.0

Table 4. Optimal policies under the three criteria (c1, c2, c3) defined in the text (actions:1="keep" , 2="replace").

Subprocess	Stage	State 1			State 2			State 3		
		c1	c2	c3	c1	c2	c3	c1	c2	c3
1	1	2	2	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2	2	2	2
1	3	2	2	2	2	2	2	2	2	2
2	1	2	2	1	1	1	1	1	1	2
2	2	2	2	2	2	2	2	2	2	2
2	3	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	1	1	1
3	2	2	2	1	1	1	1	1	1	2
3	3	2	2	2	2	2	2	2	2	2

In Table 4, optimal policies under the three criteria are shown. It appears that the same policies are optimal under the first two criteria, but under the third criterion the optimal policy differs. A more detailed example of the effect of criterion of optimality was discussed by Kristensen (1991).

In order to compare the efficiency of the hierarchic technique to the traditional policy and value iteration methods, the problem of the example was transformed to an ordinary Markov decision process and optimized by those methods. The transformed model has $3 \times 4 \times 4 = 48$ states, which is not larger than the policy iteration method may be applied without problems. In Table 5 some performance data of the three optimization techniques are compared.

The superiority of the hierarchic technique over the policy iteration method is due mainly to the time spent on solving the linear simultaneous equations of Step 2. In the hierarchic case a system of 3 equations is solved, whereas 48 equations are solved under the ordinary policy iteration method.

In this numerical example the performance of the hierarchic technique is even more superior to the value iteration method than expected from the theoretical considerations of Section 3.3. In the present case an iteration of the hierarchic model is performed even faster than one of the value iteration method applied to the same (transformed) model. The reason is that the value iteration algorithm has not been programmed in the most efficient way as defined in Section 3.3. On the contrary, the sum of Eq. (6) has been calculated over all 48 states of the transformed model. Since only 4 transition probabilities from each state are positive, the sum could be calculated only over these 4 states.

4. Uniformity: Bayesian updating

As discussed earlier, it is obvious that the traits of an animal varies no matter whether we are considering the milk yield of a dairy cow, the litter size

Table 5. The performance of the hierarchic technique compared to the policy and value iteration methods under the three criteria (c1, c2, c3) defined in the text.

	Hierarchic model			Policy iteration			Value iteration		
	c1	c2	c3	c1	c2	c3	c1	c2	c3
Number of iterations	4	3	3	3	4	3	100	100	100
Computer time, relativity	1	0.82	0.77	120	150	120	62	64	63

of a sow or almost any other trait. On the other hand, it is *not* obvious to what extent the *observed* trait Y_n at stage n is, for instant, the result of a permanent property of the animal X_1, a permanent damage caused by a previous disease X_2 or a temporary random fluctuation e_n. Most often the observed value is the result of several permanent and random effects. With Y_n, X_1, X_2 and e_n defined as above the relation might for instance be

$$Y_n = m + X_1 + aX_2 + e_n, \qquad (17)$$

where m is the expected value for an arbitrarily selected animal under the circumstances in question, and a = -1 if the animal has been suffering from the disease, and a = 0 otherwise. In this example X_1 only varies *among* animals, whereas e_n also varies *over time* for the same animal. The effect of the damage caused by the disease X_2 is in this example assumed to be constant over time when it has been "switched on". The value of X_2 is a property of the individual disease case (the "severity" of the case).

In a replacement decision it is of course important to know whether the observed value is mainly a result of a permanent effect or it is just the result of a temporary fluctuation. The problem, however, is that only the resulting value Y_n is observed, whereas the values of X_1, X_2 and e_n are unknown. On the other hand, as observations of Y_1, Y_2,... are done we are learning something about the value of the permanent effects. Furthermore, we have got a *prior* distribution of X_1 and X_2, and each time an observation is done, we are able to calculate the *posterior* distribution of X_1 and X_2 by means of the Kalman-filter theory (described for instance by Harrison and Stevens, 1976) if we assume all effects to be normally distributed.

A model as described by Eq. (17) fits very well into the structure of a hierarchic Markov process. Thus we may regard Y_n as a state variable in a subprocess, and the permanent effects X_1 and X_2 as state variables of the main process. We then face a hierarchic Markov process with *unobservable main state*. Kristensen (1993) discusses this notion in details, and it is shown that under the assumption of normally distributed effects, we only have to keep the present expected values of X_1 and X_2, the currently observed value of Y_n and (in this example) the number of stages since the animal was suffering from the disease (if it has been suffering from the disease at all). The expectations of X_1 and X_2 are sufficient to determine the current posterior distribution of the variables, because the variance is known in advance. Even though the posterior variance decreases as observations are done, the decrease does *not* depend on the *values* of Y_1, Y_2,... but only on the number of observations done.

In the study of Kristensen (1993), a more general case involving several traits each being influenced by several unobservable effects is sketched, and a numerical example involving only a single trait is given. An example concerning replacement of sows has been given by Jørgensen (1992a). It was demonstrated in both studies that the Bayesian approach in some cases may result in state space reduction without loss of information.

5. Herd restraints: Parameter iteration

One of the major difficulties identified in the introduction was *herd restraints*. All the replacement models mentioned in the previous sections have been single-component models, i.e., only one animal is considered at the same time, assuming an unlimited supply of all resources (heifers or gilts for replacement, feed, labour etc) and no production quota. In a multi-component model all animals of a herd are simultaneously considered for replacement. If all animals (components) compete for the same limited resource or quota, the replacement decision concerning an animal does not only depend on the state of that particular animal, but also on the states of the other animals (components) of the herd.

If the *only* (or at least the *most limiting*) herd restraint is a limited housing capacity, the number of animals in production is the scarce resource, and accordingly the relevant criterion of optimality is the maximization of net revenues per animal as it is expressed in the criteria (1), (2), (3) and (4). Thus the optimal replacement policy of the single component model is optimal for the multi-component model too.

If the only (or most limiting) herd restraint is a milk quota, the situation is much more complicated. Since the most limiting restriction is a fixed amount of milk to produce, the relevant criterion of optimality is now the maximization of average net revenues per kg milk yield as expressed in criterion (5), because a policy that maximizes net

revenues per kg milk will also maximize total net revenues from the herd which was assumed to be the objective of the farmer.

By following a policy which is optimal according to criterion (5) we assure at any time that the cows which produce milk in the cheapest way are kept. Thus the problem of selecting which cows to keep in the long run (and the mutual ranking of cows) is solved, but the problem of determining the optimal number of cows in production at any time is *not* solved. If for instance, it is recognized 2 months before the end of the quota year that the quota is expected to be exceeded by 10 percent, we have to choose whether to reduce the herd size or to keep the cows and pay the penalty. The problem is that both decisions will influence the possibilities of meeting the quota of the next year in an optimal way. To solve this short run quota adjustment problem we need a true multi-component model.

An other example of a herd restraint is a limited supply of heifers. If the dairy farmer only uses home-grown heifers for replacement, the actions concerning individual cows become inter-dependent, and again a multi-component model is needed in order to solve the replacement problem. Ben-Ari and Gal (1986) and later Kristensen (1992) demonstrated that the replacement problem in a dairy herd with cows and a limited supply of home grown heifers may be formulated as a Markov decision process involving millions of states. This multi-component model is based on a usual single-component Markov decision process representing one cow and its future successors. Even though the hierarchic technique has made the solution of even very large models possible, such a model is far too large for optimization in practice. Therefore, the need for an approximate method emerged, and a method called *parameter iteration* was introduced by Ben-Ari and Gal (1986).

The basic idea of the method is to approximate either the present value function $f_i(n)$ (objective function (3)) or the relative values f_i^s (objective functions (4) and (5)) by a function G involving a set of parameters $a_1,...,a_m$ to be determined in such a way that $G(i,a_1,...,a_m) \approx f_i(n)$ or $G(i,a_1,...,a_m) \approx f_i^s$.

In the implementation of Ben-Ari and Gal (1986) the parameters were determined by an iterative technique involving the solution of sets of simultaneous linear equations generated by simulation.

In a later implementation Kristensen (1992) determined the parameters by ordinary least squares regression on a simulated data set. The basic idea of the implementation is to take advantage from the fact that we are able to determine an optimal solution to the underlying (unrestricted) single-component model. If no herd restraint was present, the present value of the multi-component model would equal the sum of the present values of the individual animals determined from the underlying single-component model. Then it is argued in what way the restraint will logically reduce the (multi-component) present value, and a functional expression having the desired properties is chosen. The parameters of the function are estimated from a simulated data set, and the optimal action for a given (multi-component) state is determined as the one that maximizes the corrected present value. (A state in the multi-component model is defined from the states of the individual animals in the single-component model, and an action defines the replacement decision of each individual animal).

Ben-Ari and Gal (1986) compared the economic consequences of the resulting optimal multi-component policy to a policy defined by dairy farmers, and they showed that the policy from the parameter iteration method was better. Kristensen (1992) compared the optimal multi-component policies to policies from usual single-component models in extensive stochastic simulations and showed that the multi-component policies were superior in situations with shortage of heifers.

The parameter iteration method has been applied under a limited supply of heifers. It seems to be realistic to expect, that the method and the basic principles of Kristensen (1992) may be used under other kinds of herd restraints as for instance the short time adjustment to a milk quota as mentioned above.

6. General discussion

In the introduction, the main difficulties of the animal replacement problem were identified as *variability* in traits, *cyclic production*, *uniformity* (the traits are difficult to define and measure) and *herd restraints*. We are now able to conclude that the difficulties of variability and the cyclic production are directly solved by the application of Markov decision programming, but when the variability of several traits are included we face a problem of *di-*

mensionality. The formulation of the notion of a *hierarchic Markov process* contributed to the solution of the dimensionality problem, but did not solve it. The upper limit of number of states to be included has been raised considerably, but not eliminated.

This is for instance clearly illustrated when we formulate *multi-component* herd models in order to deal with herd restraints. In that case we still have to use approximate methods to determine an "optimal" replacement policy. On the other hand it has been demonstrated by Kristensen (1992) that the *parameter iteration* method applied to a multi-component herd model (even though it is only approximate) is able to improve the total net revenue compared to the application of a usual single-component (animal) model in a situation with shortage of heifers. The parameter iteration method is an important contribution to the problem of determining optimal replacement policies under herd restraints.

In other situations with a limiting herd restraint it may be relevant to use an alternative criterion of optimality maximizing average net revenue per unit of the limiting factor. This method has been successfully applied in a situation with milk production under a limiting quota.

Recent results have also contributed to the solution of the uniformity problem. The Bayesian updating technique described in Section 4 seems to be a promising approach, but it has not yet been tested on real data. It might be a solution to the problem of including animal health as a trait to be considered. The problem of including diseases in the state space has never been solved, but at present Houben et al. (1992) are working on it. As concerns other traits such as litter size or milk yield the Bayesian approach may in some cases result in a reduction of the state space without loss of information (Jørgensen, 1992a; Kristensen, 1993). Thus it contributes indirectly to the solution of the dimensionality problem.

References

Ben-Ari, Y. and Gal, S. 1986. Optimal replacement policy for multicomponent systems: An application to a dairy herd. *European Journal of Operational Research* 23, 213-221.

Ben-Ari, Y., Amir, I. and Sharar, S. 1983. Operational replacement decision model for dairy herds. *Journal of Dairy Science* 66, 1747-1759.

Giaever, H. B. 1966. *Optimal dairy cow replacement policies*. Ph. D. dissertation. Ann Ar-bor, Michigan: University Microfilms.

Harrison, P. J. and Stevens, C. F. 1976. Bayesian forecasting. *Journal of the Royal Statistical Society B* 38, 205-247.

Houben, E. H. P., Dijkhuizen, A. A. and Huirne, R. B. M. 1992. Knowledge based support of treatment and replacement decisions in dairy cows with special attention to mastitis. *Proceedings of the SKBS/MPP Symposium '92*, March 11, Zeist, The Netherlands.

Howard, R. A. 1960. *Dynamic programming and Markov processes*. Cambridge, Massachusetts: The M.I.T. Press.

Howard, R. A. 1971. *Dynamic probabilistic systems. Volume II: Semi-Markov and decision processes*. New York: John Wiley & Sons, Inc.

Huirne, R. B. M., Hendriks, Th. H. B., Dijkhuizen, A. A. and Giesen, G. W. J. 1988. The economic optimization of sow replacement decisions by stochastic dynamic programming. *Journal of Agricultural Economics* 39, 426-438.

Huirne, R. B. M., van Beek, P., Hendriks, Th. H. B. and Dijkhuizen, A. A. 1992. An application of stochastic dynamic programming to support sow replacement decisions. *European Journal of Operational Research*. In press.

Jenkins, K. B. and Halter, A. N. 1963. A multistage stochastic replacement decision model: Application to replacement of dairy cows. *Technical Bulletin* 67, Agr. Exp. Sta., Oregon State University.

Jewell, W. 1963. Markov renewal programming I and II. *Operations Research* 11, 938-971.

Jørgensen, E. 1992a. Sow replacement: Reduction of state space in Dynamic Programming model and evaluation of benefit from using the model. Submitted for journal publication.

Jørgensen, E. 1992b. The influence of weighing precision on delivery decisions in slaughter pig production. *Acta Agriculturæ Scandinavica, Section A, Animal Science*. In press.

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.

Kennedy, J. O. S. 1986. *Dynamic programming. Applications to agriculture and natural resources*. London and New York: Elsevier applied science publishers.

Killen, L. and Kearney, B. 1978. Optimal dairy cow replacement policy. *Irish Journal of Agricultural Economics and Rural Sociology* 7, 33-40.

Kristensen, A. R. 1987. Optimal replacement and ranking of dairy cows determined by a hierarchic Markov process. *Livestock Production Science* 16, 131-144.

Kristensen, A. R. 1988. Hierarchic Markov processes and their applications in replacement models. *European Journal of Operational Research* 35, 207-215.

Kristensen, A. R. 1989. Optimal replacement and ranking of dairy cows under milk quotas. *Acta Agriculturæ Scandinavica* 39, 311-318.

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. 1992. Optimal replacement in the dairy herd: A multi-component system. *Agricultural Systems* 39, 1-24.

Kristensen, A. R. 1993. Bayesian updating in hierarchic Markov processes applied to the animal replacement problem. *European Review of Agricultural Economics* 20, 223-239.

Kristensen, A. R. and Thysen, I. 1991. Economic value of culling information in the presence and absence of a milk quota. *Acta Agriculturæ Scandinavica* 41, 129-135.

Kristensen, A. R. and Østergaard, V. 1982. Optimalt udskiftningstidspunkt for malkekoen fastlagt ved en stokastisk model. (In Danish with English summary). *Beretning fra Statens Husdyrbrugsforsøg 533.*

McArthur, A. T. G. 1973. Application of dynamic programming to the culling decision in dairy cattle. *Proceedings of the New Zealand Society of Animal Production* 33, 141-147.

Reenberg, H. 1979. Udskiftning af malkekøer. En stokastisk udskiftningsmodel. (In Danish with English summary). *Memorandum 6*, Jordbrugsøkonomisk Institut.

Ross, S. M. 1970. *Applied probability models with optimization applications*. San Francisco, California: Holden-Day.

Smith, B. J. 1971. The dairy cow replacement problem. An application of dynamic programming. *Bulletin 745*. Florida Agricultural Experiment Station. Gainesville, Florida.

Stewart, H. M., Burnside, E. B. and Pfeiffer, W. C. 1978. Optimal culling strategies for dairy cows of different breeds. *Journal of Dairy Science* 61, 1605-1615.

Stewart, H. M., Burnside, E. B., Wilton, J. W. and Pfeiffer, W. C. 1977. A dynamic programming approach to culling decisions in commercial dairy herds. *Journal of Dairy Science* 60, 602-617.

van Arendonk, J. A. M. 1984. Studies on the replacement policies in dairy cattle. I. Evaluation of techniques to determine the optimum time for replacement and to rank cows on future profitability. *Zeitschrift für Tierzüchtung und Züchtungsbiologi* 101, 330-340.

van Arendonk, J. A. M. 1985. Studies on the replacement policies in dairy cattle. II. Optimum policy and influence of changes in production and prices. *Livestock Production Science* 13, 101-121.

van Arendonk, J. A. M. 1986. Studies on the replacement policies in dairy cattle. IV. Influence of seasonal variation in performance and prices. *Livestock Production Science* 14, 15-28.

van Arendonk, J. A. M. 1988. Management guides for insemination and replacement decisions. *Journal of Dairy Science* 71, 1050-1057.

van Arendonk, J. A. M. and Dijkhuizen, A. A. 1985. Studies on the replacement policies in dairy cattle. III. Influence of variation in reproduction and production. *Livestock Production Science* 13, 333-349.

van der Wal, J. and Wessels, J. 1985. Markov decision processes. *Statistica Neerlandica 39*, 219-233.

White, C. C. III and White, D. J. 1989. Markov decision processes. *European Journal of Operational Research 39*, 1-16.