Mathematical modelling of insect oviposition behaviour

Baring-Gould, Simone

Baring-Gould, Simone (1998) *Mathematical modelling of insect oviposition behaviour*, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/5040/

Abstract

Simone Baring-Gould
Mathematical Modelling of Insect Oviposition Behaviour
M.Sc., 1998

This thesis is concerned with various aspects of insect oviposition behaviour.

In the first chapter published mathematical models developed to understand optimal insect oviposition behaviour are reviewed. In these models it is assumed that selection favours females that maximize their offspring's total reproductive success.

In the second chapter a different approach to the optimization problem is presented. It is shown that the quantity that is maximized in the models that were discussed in the review is not well defined. It is suggested that instead the total expected resource gain that can be acquired by a female's offspring should be used as a fitness measure. The main reason for this is that if fitness is defined as the ability to pass genes on to all future generations, maximizing the fitness measure used in the existing models would not completely resolve the recursive nature of this definition.

The third chapter investigates the effects of density-dependent fecundity on population size. It is assumed that females lay only one single clutch and that the size of the clutch is directly related to the female's fecundity. An iterative model is derived to calculate variation in population size. An analysis of the model and subsequent simulation predict that low levels of competition among larvae is likely to cause chaotic behaviour and overpopulation of the environment whereas high competition is likely to have a stabilizing effect on population size.

A fourth chapter briefly summarizes an experiment conducted on *Pieris brassicae* to measure variation in egg size and to estimate larval survival rates.

Mathematical Modelling
of
Insect Oviposition Behaviour

Simone Baring-Gould

M.Sc.
University of Durham
Department of Biology

1998

Contents

Chapter 1

Literature review: optimization of insect oviposition behaviour

1.1 Introduction

In this chapter I will review published mathematical optimization models that have been developed to explain and predict insect oviposition behaviour. The models reviewed here are all based on the assumption that a female should optimize her fitness by producing as many surviving offspring as possible while simultaneously optimizing the offspring's reproductive prospects (see Parker and Maynard Smith (1990)). They have been shown to be relatively realistic and mathematically tractable choices.

Section 1.2 will discuss the most basic model where fitness is maximized for a single clutch. The following two sections present improvements of this model. Section 1.3 will discuss maximization of the rate at which fitness is gained and section 1.4 will discuss a model which maximizes fitness over a female's whole lifetime by including the female's mortality risk.

Section 1.5 presents models where fitness maximization is studied from the evolutionary viewpoint.

Finally a dynamic programming approach to fitness maximization will be presented in section 1.6.

1.2 Maximizing fitness per clutch

The problem of optimizing fitness per clutch was first addressed by Lack in 1947. The resulting optimal clutch size is known as the Lack solution clutch size and will be denoted by n^* throughout this chapter. Whereas Lack only discussed variation in clutch size, Parker and Begon later included variation in egg size (see Parker and Begon (1986) and Begon and Parker (1986)).

Let s denote a function that describes the per capita fitness of the offspring of a clutch depending on the size of the clutch. Per capita fitness, is defined as the product of survival probability and future reproductive success.

Multiplying $s(n)$ by n gives the combined fitness of all surviving offspring from a clutch of size n. Denote the fitness function by w. Then

$$w(n) = ns(n).$$

Parker and Begon (1986) suggested that the fitness obtained through a single clutch should also depend on the size of the eggs, which they denoted by m. They assumed that egg size is constant within clutches and varies only among clutches and that the per capita fitness of the offspring of a clutch depends on n and m separately. They introduced a second function f, which depends only on m and expressed fitness as:

$$w(n,m) = ns(n)f(m).$$

Remark:

For a variable to maximize a function, two conditions have to be satisfied: (1) the first derivative has to be equal to zero and (2) the second derivative has to be negative. In most of the quoted papers the second condition has not been discussed since the equations tend to become too complicated to lead to any further conclusions about the solution of the optimization problem. Therefore, in this chapter only the first condition will be considered.

In the following section several different choices for the functions s and f will be discussed.

1.2.1 Functional representations of per capita fitness

Functions that model the per capita fitness - clutch size relationship.

The following functions have been argued to be realistic choices for s in a variety of different circumstances.

Define

$$s_1(n) := \begin{cases} b\left(1-\frac{n}{v}\right)^r & \text{if} \quad 0 < n < v \\ 0 & \text{if} \quad n \geq v \end{cases}, \tag{1.1}$$

where $b, v > 0$ and $r \neq 0$.

Define

$$s_2(n) := be^{-\sigma n}, \tag{1.2}$$

where $b, \sigma > 0$, and

$$s_3(n) := b(1+\alpha n)^{-\tau}, \tag{1.3}$$

where $b, \alpha, \tau > 0$.

The parameter r in the function s_1 controls the severity of the competition among the offspring of a clutch. For negative r s_1 describes "contest" competition (see Hassell (1978)). s_1 with $r = 1$ was used by Suzuki and Iwasa (1980) and by Ives (1991) and s_1 with $r = 2$ was used by Parker and Courtney (1984). Parker and Courtney also used $s(n) = 1 - B - cn^2$ where $0 < B, c < 1$.

The second function, s_2, was shown by Waage and Godfray (1985) to give good empirical fit to competition among parasitic wasps. s_2 and s_3, were used by Ives (1991) as descriptions for competition.

Note that s_1, s_2 and s_3 are all decreasing functions of clutch size. It has been shown in experiments that in some cases per capita fitness of the offspring increases initially up to a certain clutch size and then decreases. This is referred to as Allee's effect (see Allee *et al.* (1949)).

Functions that model the per capita fitness - egg size relationship.

The function f describes the dependency of the per capita fitness of the offspring on the amount of resources provided in the egg by its mother. It should have the following two properties (see Parker and Begon (1986)):

- The egg size m should not decrease below a certain minimum egg size m_0 which represents the minimum amount of resources that are necessary for the offspring to develop.
- As m increases f should increase monotonically and converge to some limiting value α above which an increase in egg size does not result in any further increase of the offspring's viability.

Parker and Begon (1986) use

$$f(m) := 1 - e^{-(m-m_0)}$$

for their models.

In the next two sections I will first discuss the case where the fitness of the clutch depends only on the clutch size n. After that I will discuss the case where fitness also depends on the egg size m.

1.2.2 Lack solution clutch size

If egg size is constant and the per capita fitness of the offspring is assumed to depend only on the clutch size n, the fitness of the clutch is given by

$$w(n) := n \cdot s(n). \tag{1.4}$$

For n^* to maximize w, the following condition has to be satisfied:

$$w'(n^*) \;=\; n^* s'(n^*) + s(n^*) \;=\; 0$$

or equivalently

$$n^* \frac{s'(n^*)}{s(n^*)} + 1 = 0 \tag{1.5}$$

The Lack solution clutch sizes, n_1^*, n_2^* and n_3^*, for the survival functions described in section 1.2.1 are obtained from (1.5):

$$\begin{aligned} n_1^* &= \frac{v}{(r+1)} \\ n_2^* &= \frac{1}{\sigma} \\ n_3^* &= \frac{1}{\alpha(\tau - 1)}. \end{aligned}$$

1.2.3 Lack solution clutch size and variation in egg size

In this section I will briefly discuss the case where variation of egg size is included when fitness is maximized per clutch. This will make it easier to see how the equations derived by Parker and Begon in [Parker/Begon 86] and [Begon/Parker 86] are related to each other and to the other models discussed in this chapter.

If per capita fitness of an offspring depends on the size of its egg in the way described at the beginning of section 1.2, i.e. if

$$w(n,m) = ns(n)f(m), \tag{1.6}$$

the optimal egg and clutch size will depend on the total amount of resources that are invested into the clutch. Denote this quantity by M and let M be measured in the same units as m. Then M, n and m are related via $M = nm$ and optimizing fitness is equivalent to finding the optimal way of dividing up the resources that are available to lay the clutch.

To find the optimal clutch size and egg size, w will be maximized using the Lagrange multiplier method subject to the constraint $M = nm$. This leads to the equations

$$\begin{aligned} \frac{\partial}{\partial n}(w(n,m) + \lambda(M - nm)) &= 0 \\ &\Leftrightarrow && (1.7) \\ f(m)(ns'(n) + s(n)) &= \lambda m && (1.8) \end{aligned}$$

and

$$\begin{aligned} \frac{\partial}{\partial m}(w(n,m) + \lambda(M - nm)) &= 0 \\ &\Leftrightarrow && (1.9) \\ ns(n)f'(m) &= \lambda n && (1.10) \end{aligned}$$

Equation (1.8) and (1.10) combine to

$$\frac{f(m)}{m}(ns'(n) + s(n)) = s(n)f'(m)$$

or equivalently

$$n\frac{s'(n)}{s(n)} + 1 = m\frac{f'(m)}{f(m)}. \tag{1.11}$$

The optimal clutch size and egg size can be computed from (1.11) and from $M = nm$.

1.3 Maximizing the rate of fitness gain

The model described in the previous section does not take into account the time that a female spends ovipositing, searching for oviposition sites, foraging for food etc. It predicts only that if a female were to lay just one clutch in her life she should produce the Lack clutch size. The models reviewed in this section improve this by maximizing the rate at which fitness is gained.

The two classical foraging models (see Krebs and McCleery (1984) and Stephens and Krebs (1986)), the classical model of prey choice and the marginal value theorem, both aim to maximize the long term rate at which energy is accumulated through foraging. They have been subsequently applied to the problem of finding optimal oviposition strategies.

1.3.1 The marginal value theorem applied to oviposition behaviour

Charnov *et al.* (1976) developed the marginal value theorem to model resource depression in foraging processes. It maximizes the rate at which energy is accumulated through foraging given a fixed travel time between patches and proves the existence of a threshold value above which the insect should move on to a not yet depleted food patch. It predicts that the insect should accumulate less energy per patch than it would if it were to visit only one patch.

Similarly to resource depletion, adding eggs to a clutch yields diminishing returns. The two following models solve the resulting trade-off by applying to it the marginal value theorem. Analogous to the prediction of the original version, clutch size is predicted to be generally smaller than the Lack clutch size. Optimizing population growth rate Charnov and Krebs (1974) show in a similar way that the optimal clutch for a whole population should in general be smaller than the most productive clutch.

The first of the following two models was developed independently by Parker and Courtney (1984) and by Skinner (1985). Skinner optimized oviposition time given a fixed searching time and Parker and Courtney optimized reproduction time, again, given a fixed searching time. The second was developed by Parker and Begon (1986) and includes variation in egg size. They optimized foraging time given a fixed searching time and neglecting oviposition time.

Optimizing reproduction time

Assume that the survival of the offspring does not depend on external factors, such as the value of the patch on which the eggs are oviposited, but solely on the clutch size. The rate of fitness gain can be defined as

$$\begin{aligned} G &:= \frac{\text{fitness gained through a particular clutch}}{\text{duration of one oviposition period}} \\ &= \frac{w}{T}. \end{aligned} \tag{1.12}$$

Skinner (1985) expresses G in terms of the oviposition time t. To do this he assumes that the female lays eggs at a fixed rate R, i.e. that $n(t) := Rt$.

Parker and Courtney (1984) express G and the oviposition time t in terms of n. This allows them to include other egg-dependent time factors. According to their approach, G can be written as

$$G(n) = \frac{ns(n)}{d + t(n)},$$

where t denotes the reproduction time , i.e. the time spent producing the clutch which can consist of several egg-dependent components such as the time taken to obtain the resources that are needed to mature the eggs and the time taken to oviposit them. d denotes the egg-independent time spent between the patches and is assumed to be constant.

Differentiating G w.r.t. n gives

$$G'(n) = \frac{ns'(n) + s(n)}{d + t(n)} - ns(n)\frac{t'(n)}{(d + t(n))^2}.$$

The condition for n to maximize G is $G'(n) = 0$, i.e.

$$\frac{ns'(n) + s(n)}{s(n)} = \frac{nt'(n)}{d + t(n)}$$

or equivalently

$$n\frac{s'(n)}{s(n)} + 1 = \frac{nt'(n)}{d + t(n)}. \tag{1.13}$$

If eggs are laid at a fixed rate, i.e. if $n = Rt$, one obtains

$$n\frac{s'(n)}{s(n)} + 1 = \frac{t}{d + t}. \tag{1.14}$$

Optimizing foraging time

Parker and Begon (1986) discuss variation in clutch size and egg size using the above approach of maximizing the rate of fitness gain. Egg size is assumed to be constant within clutches and to vary only among clutches. As fitness functions $w(n,m) := nf(m)$ and $w(n,m) := ns(n)f(m)$ are used.

They take a slightly different viewpoint compared to Parker and Courtney (1986) and Skinner (1985). Again the duration of the oviposition period is split up into egg-dependent time cost t and egg-independent time cost d. d is the searching time. t is the foraging time and to be optimized. It determines directly the amount of reproductive resources $M(t)$ that the female will have available for the current oviposition period. The time spent ovipositing is assumed to be negligible.

Egg size and clutch size are related via $M = nm$. Choosing t and m as variables clutch size can be expressed as $n(t,m) = \frac{M(t)}{m}$.

For $w(n,m) := nf(m)$ the rate of fitness gain is then given by

$$G(t,m) = \frac{M(t)}{m} f(m) \frac{1}{t+d}.$$

For $w(n,m) := ns(n)f(m)$ we have

$$G(t,m) = \frac{M(t)}{m} s(n(t,m)) f(m) \frac{1}{t+d}.$$

Using Lagrange's method of optimization subject to the constraint $M = nm$ leads to the conditions

$$1 = m\frac{f'(m)}{f(m)} = \frac{1}{t+d}\frac{M(t)}{M'(t)} \quad \text{and} \quad n(t,m) = \frac{M(t)}{m}$$

in the first case, and

$$n\frac{s'(n)}{s(n)} + 1 = m\frac{f'(m)}{f(m)} = \frac{1}{t+d}\frac{M(t)}{M'(t)} \quad \text{and} \quad n(t,m) = \frac{M(t)}{m}$$

in the second case.

Note that in the second case the left part of the first equation is equivalent to equation (1.11). This means that the optimal arrangement of the resources $M(t)$ is the same as if fitness were maximized per clutch.

For an application of the marginal value theorem to strategies for optimal host exploitation in parasitoids also see Charnov and Skinner (1988).

1.3.2 The classical model of prey choice applied to oviposition behaviour

The classical model of prey choice maximizes the rate of energy intake gained from different types of prey. Iwasa *et al.* (1984) used this to model oviposition behaviour of parasitoids. The rate at which fitness is gained is given by

$$r(x_1, ..., x_k) := \frac{\sum_{i=1}^{k} G_i(x_i)\lambda_i}{1 + \sum_{i=1}^{k} H_i(x_i)\lambda_i},$$

where λ_i is the frequency of encounter with a host of type i, G_i describes the reproductive success obtained when eggs are laid on a host of type i, H_i determines the handling time from the number of eggs laid on a host of type i and x_i is the clutch size the female should produce when ovipositing on a host of type i. This equation simplifies if eggs are laid singly. Notice also that Iwasa *et al.* (1984) do not consider cases where a female parasitoid can encounter previously attacked hosts.

For solitary parasitoids, i.e. if eggs are laid singly, Iwasa *et al.* (1984) find that the range of host type utilization should be narrower in an environment with a higher density of hosts. For gregarious parasitoids, i.e. if a varying number of eggs is oviposited on a host, they find that the threshold value which determines whether the female should or should not oviposit on a host, corresponds to the one predicted by the marginal value theorem for a predator's *optimal patch use* problem (see Charnov (1976)) and is the same for all hosts.

1.4 Maximizing the expected total fitness gain

The models discussed in the previous section do not take into account any mortality risk that the female might suffer from during her lifetime. Also, as Iwasa *et al.* (1984) pointed out, they are not suitable where time and number of eggs are limited since they focus on the average fitness gain. The models reviewed in this section maximize the expected total fitness gain of a female, i.e. the sum over the fitness gains obtained from all clutches laid during the female's lifetime including as weights the female's age-dependent mortality risk.

Iwasa *et al.* (1984) and Mangel (1987) used dynamic programming to optimize a female's expected total fitness gain. Their approach will be presented in section 1.6.

An expression for the expected total fitness gain is obtained by weighting the fitness gains of each clutch with the probability that the clutch is going to be laid and adding them together. This model has been discussed by Parker and

Courtney (1984) and by Begon and Parker (1986).

Let p_i be the probability of surviving to lay the i^{th} clutch and assume that this is not dependent on prior reproductive effort. Furthermore, let n_i denote the clutch size of the i^{th} clutch. Then the expected total fitness gain of a female is given by

$$w(n_1, ..., n_k) = \sum_{i=1}^{k} p_i \cdot n_i \cdot s(n_i) \tag{1.15}$$

where k is the maximum number of clutches that can be laid. The amount of resources the female has available for reproduction are assumed to be limited. This means that the number of clutches is limited and hence determines k.

This model was studied by Parker and Courtney (1984) for a fixed risk of death when searching for a patch and both for constant and for varying clutch sizes. Begon and Parker (1986) studied this case for varying mortality risks and for varying clutch sizes and egg sizes.

Fixed risk of death

Parker and Courtney (1984) consider a situation where the female sustains a fixed risk of death, p, due to searching for a larval food source, so that the probability of survival after i periods of searching is given by p^i.

If all clutch sizes are the same, i.e. if $n_1 = n_2 = ... = n_k =: n$, and if it is assumed that the female can only lay a fixed number of eggs, E say, the expected total fitness gain of the female is given by

$$\begin{aligned} w(n) &= n \cdot s(n) \cdot (1 + p^1 + p^2 + p^3 + ... + p^k) \\ &= n \cdot s(n) \cdot \frac{p - p^{k+1}}{1-p}, \end{aligned}$$

where

$$k = \left[\frac{E}{n}\right].$$

Computing the optimal clutch size in the usual way gives

$$n = -\frac{s(n)}{s'(n)} \left(1 + \frac{E \cdot ln(p) \cdot p^{E/n}}{n(1 - p^{E/n})}\right).$$

Parker and Courtney (1984) also mention that if k is independent of n, the constant $\frac{p-p^{k+1}}{1-p}$ will be lost when w is differentiated and set to zero. Thus, the optimal clutch size will be the Lack clutch size. As examples they list the case where k is determined by the termination of the breeding season and the case where the females die after reaching a fixed age.

Remark:

Suppose that the oviposition period is terminated by the end of the breeding season or by the death of the female at a fixed age. Denote by k_{max} the maximum number of clutches that the female can lay in the limited time span that is available to her for oviposition. Denote by k_{min} the minimum number of clutches that she would lay without fixed time limit. This is given by $k_{min} = E/n^*$, where E are her reproductive reserves and n^* is the Lack solution clutch size.

If $k_{max} \leq k_{min}$, the number of clutches laid by the female is independent of her reproductive reserves and the clutch sizes and as Parker and Courtney (1984) conclude the female should produce Lack clutches. If, however, $k_{max} > k_{min}$, the maximum number of clutches laid by the female is given by $k = min\{\frac{E}{n}, k_{max}\}$ and hence depends on n.

Thus, a female should produce Lack clutches only if the oviposition period is so short that the female is lacking time rather than lacking reproductive reserves.

Variable mortality risk, clutch size and egg size

Begon and Parker (1986) study this model in the case where both, egg size and clutch size, vary. The expected total fitness gain is then given by

$$w(n_1, ..., n_k, m_1, ..., m_k) = \sum_{i=1}^{k} p_i \cdot n_i \cdot s(n_i) \cdot f(m_i), \tag{1.16}$$

where it is assumed that the female has only a limited amount of resources M available. Resources and egg size can be measured in the same units and hence M can be written as $M = \sum_{i=1}^{k} n_i \cdot m_i$.

Begon and Parker (1986) use Lagrange's method to discuss the optimal strategy $(n_1, ..., n_k, m_1, ..., m_k)$. They find that the optimal arrangement of egg size and clutch size is independent of adult mortality and that solving the above problem is equivalent to optimizing the way the resources $M_i := n_i \cdot m_i$ are allocated to the clutches which the female will lay during her lifetime. One can in fact show that if mortality risks decline the resources invested into the clutches should decline, too. For a detailed explanation and for a further discussion of the model see chapter 2.

1.5 Clutch size when many females oviposit on a patch: evolutionarily stable clutch size

Fitness maximization per clutch as described in section 1.2 can be extended to the case where more than one female oviposits on a patch. The fitness function w of a clutch is then given by

$$w(n) = n \cdot s(n + e),$$

where e is the number of eggs laid by all other females. In a similar fashion the egg sizes produced by other females can be included.

Ives (1991) uses this to calculate the evolutionarily stable clutch size (ESS). Suppose that F other females oviposit on the patch and that each of them lays $\bar{n}$ eggs. The fitness gained by a mutant female that lays n eggs is given by

$$w(n) = n \cdot s(n + F\bar{n}).$$

The evolutionarily stable clutch size is obtained by differentiating w as usual and by setting the mutant clutch size n equal to the population clutch size $\bar{n}$ in the resulting equations. This gives the conditions

$$\begin{aligned} & w'(n^*) &=& \; n^* s'(n^* + F\bar{n}) + s(n^* + F\bar{n}) \quad = 0 \\ \text{and} \quad & n^* &=& \; \bar{n}. \end{aligned}$$

Ives calculates the evolutionarily stable clutch size for different kinds of survival probability functions s. For s_1 with $r = 1$ and for s_2 and s_3 he obtains

$$\begin{aligned} \hat{n}_1 &= \frac{v}{F+1} \\ \hat{n}_2 &= \frac{1}{\sigma} \\ \hat{n}_3 &= \frac{1}{\alpha(\tau - F)} \end{aligned}$$

He discusses three modifications of the basic model:

(1) the number of other females encountered on oviposition patches varies according to some probability distribution,

(2) fitness is maximized per unit time,

(3) combines (1) and (2) and studies this for females that can assess whether other females have previously oviposited on a patch.

In their paper on optimal egg size and clutch size Parker and Begon (1986) derive a model to calculate evolutionarily stable egg size and clutch size including the relative competitive success of one egg against another.

Optimizing egg size and clutch size per clutch showed that for a fixed amount of reserves M the optimal arrangement of egg size and clutch size is determined by equation (1.11). This means that one can write the fitness function w (see (1.6) as

$$w(m) = \frac{M}{m} s(n) f(m),$$

where $n = \frac{M}{m}$.

Parker and Begon (1986) introduce into w a third factor c, the relative competitive success of one egg against another. c takes as argument the ratio $\frac{m}{\bar{m}}$ of the mutant egg size to the average egg size produced by the other females and returns a value greater than 1 if $m > \bar{m}$, smaller than 1 if $m < \bar{m}$ and 1 if $m = \bar{m}$.

If F females oviposit on a patch the clutch fitness is given by

$$w(m) = \frac{M}{m} s(N) f(m) c(\frac{m}{\bar{m}}),$$

where $N := \frac{M}{m} + \frac{M}{\bar{m}}(F-1)$ is the total number of eggs and $\bar{m} := \frac{FM}{N}$ the average egg size.

The evolutionarily stable egg size m^* is obtained in the usual way. The result is

$$\frac{1}{F} N^* \frac{s'(N^*)}{s(N^*)} + 1 - \frac{F-1}{F} \frac{c'(1)}{c(1)} = m^* \frac{f'(m^*)}{f(m^*)}, \qquad (1.17)$$

where $N^* := \frac{M}{m^*} + \frac{M}{m^*}(F-1)$.

Parker and Begon (1986) generalize this by allowing females to have different amounts of resources. They assume that at each oviposition site there are always F females having $M_1, M_2, ..., M_F$ amounts of resources.

For a female playing $m_i \neq \hat{m}_i$ the clutch fitness is given by

$$w(m_i, \hat{m}_1, ..., \hat{m}_F) = \frac{M_i}{m_i} s(N) f(m_i) c(\frac{m_i}{\bar{m}}),$$

where $N := \sum_{j=1}^{F} \frac{M_i}{\hat{m}_i}$ is the total number of eggs and $\bar{m} := \frac{F\bar{M}}{N}$ the average egg size ($\bar{M} = \frac{1}{F}\sum_{j=1}^{F} M_i$).

The evolutionarily stable egg size m_i^* is obtained in the usual way. The result is

$$\frac{M_i}{m_i^*}\frac{s'(N^*)}{s(N^*)} + 1 - \frac{F\frac{m_i^*}{\bar{m}} - \frac{M_i}{\bar{M}}}{F}\frac{c'(\frac{m_i^*}{\bar{m}})}{c(\frac{m_i^*}{\bar{m}})} = m_i^*\frac{f'(m_i^*)}{f(m_i^*)}, \tag{1.18}$$

where $N^* := \sum_{j=1}^{F} \frac{M_i}{m_i^*}$.

Note that setting $m = m_i^* = \bar{m}$ and $M = M_i = \bar{M}$ in equation (1.18) gives equation (1.17).

1.6 Dynamic programming

First developed by Bellman in 1957, dynamic programming provides a method of finding the optimal path through a graph which represents many possible strategies. It provides a way of solving optimization problems which have the following underlying properties:

- they can be divided into separate stages. So-called state variables describe the system's state at a particular stage. At each stage a decision is taken based on the state the system is in. The state variables must contain all the information needed to make that decision. This is referred to as *state separation property* or *Markovian state property*.

- the quantity that is to be optimized (i.e. maximized or minimized) depends on the decisions made at each stage. It can be represented by a function f, the so-called objective function, which has the following property

  $$f(x_1, ..., x_k) = \bigotimes_{j=1}^{k} f_j(x_j),$$

  where $x_1, ..., x_k$ are the decisions, the f_j's are functions and $\bigotimes$ is some operator. This property is referred to as *separability of the objective function.* The f_j's are also referred to as *stage returns*.

These two properties invoke

Bellman's Principal of Optimality:

An optimal strategy has the property that whatever the initial state and the initial decision are, the remaining decisions must constitute an optimal strategy with respect to the state which results from the initial decision.

This condition leads to recurrence relations on which the method of dynamic programming is based.

The optimization problem given by (1.16) is an example of a dynamic programming problem and we will refer to it in chapter 2.

A dynamic programming problem is solved backwards. For all possible states that the system can attain in the last stage, the stage return is optimized over all possible decisions. In the last but one stage a decision determines the state which the system will attain in the last stage. For this state the optimal decision has just been calculated. So, in the last but one stage the cumulative return from the last two stages is optimized over all possible decisions – this is where one needs the separability of the objective function. The iteration now continues backwards until it reaches the first stage where the system is in the initial state. Optimizing as before gives the optimal first decision. From this the complete optimal strategy can now be computed, this time iterating forwards until reaching the last stage.

Iwasa *et al.* (1984) extended the classical model of prey choice by including mortality risks that a female parasitoid suffers from when searching for hosts and when handling them. They derived an iterative equation for the expected reproductive success obtained by a searching female parasitoid through all future ovipositions. They solved this equation using the method of dynamic programming. Their model leads to the following three predictions: (1) if the mortality during handling a host is common between hosts the host range should be wider for a greater number of eggs in the female's body; (2) mortality during handling is more important when the number of available eggs is large and hence a female with a larger number of eggs available should avoid high mortality hosts; (3) the number of eggs laid per host should depend on the mortality and the number of available eggs.

Oviposition behaviour and dynamic models were also studied by Mangel (see Mangel (1987) and Mangel and Clark (1988)). He discusses models where a female insect can encounter different types of hosts with different probabilities. In the case where all eggs are assumed to be already mature at the start of the process the predictions are that older insects should be less selective about host types, clutches should generally be smaller than the Lack clutch size unless time is limited and larger clutches should occur more frequently as mortality increases. A second case discusses the situation where females have an essentially unlimited number of oocytes which have to be matured and for which resources have to be gathered. Here females should spend less time foraging for food and more time searching for oviposition sites as they get older. Several other models are derived including one where oocytes are limited and two modifications of the first model, one where stages are continuous and one where handling time is included.

1.7 Other models

To complete the review of published optimization models a model developed by Parker and Courtney (1984), which did not find a place in the previous sections, and Godfray and Ives (1988) stochastic approach to oviposition behaviour will be briefly summarized.

Parker and Courtney model mortality anticipation, deriving an expression for the number of food units, K, provided by a patch in terms of the food consumption $f(t)$ and larval survival probability $s(t)$ at age t:

$$K := n_k \int_0^T f(t)s(t)dt,$$

where n_k is the number of eggs that will consume exactly K food units by the pupation time T. It is assumed that K is constant and that the probability of death due to starvation (different from s) is 1 if K is exceeded by the offspring's food consumption and 0 otherwise. Also it is assumed that only one female oviposits in a patch and that there is a cost to finding another patch, so that it should be optimal for a female to lay just enough eggs to consume the value K by pupation time T. The model predicts that: (1) species with shorter development times should lay bigger clutches; (2) large resources favour large clutches; (3) clutch size should be related to larval growth rates and death rates.

Godfray and Ives investigate the effects of introducing behavioural and environmental stochasticity into some of the models reviewed in the previous sections. They first analyze the effect of stochastically varying patch quality. They then examine optimal strategies of females that cannot precisely control their clutch sizes. Finally, they study the case where the trade-off between clutch size and number of clutches is governed by a stochastic variable. They find that introducing variability in the parameters of simple optimization models can affect, sometimes markedly, the predicted optimal oviposition behaviour and hence that stochasticity needs to be considered when studying insect oviposition behaviour.

For a study of work on invertebrate clutch size see Godfray (1987) who reviewed predictions of optimization models for clutch size and discussed quantitative tests of these predictions in relation to comparative studies of invertebrate clutch size.

1.8 Discussion

The models reviewed in this chapter optimize a measure for the lifetime fitness gain that a female can obtain through adopting a particular oviposition behaviour. In all models the fitness gain is calculated from the fitness gains obtained from the individual clutches that a female produces. As measure of the fitness of a single clutch the quantity

$$\left(\begin{array}{l}\text{number of eggs}\\ \text{in the clutch}\end{array}\right) \cdot \left(\begin{array}{l}\text{survival probability of}\\ \text{a larva of the clutch}\end{array}\right) \cdot \left(\begin{array}{l}\text{fecundity of a larva}\\ \text{of the clutch}\end{array}\right)$$

is used. The second and third factor of this product are represented by the function s. I believe that a clear distinction should be made between these two factors because they are responsible for the trade-off between producing many possibly weak offspring and producing very successful but only few offspring. Instead of representing them by one function only, I believe, they should be discussed separately and mathematically represented by two different functions. I also believe that the third factor in the above product should not be the fecundity of a larva of that clutch but the amount of resources that the larva can accumulate given the size of the clutch. The reasons for this and a model in which these ideas are implemented will be discussed in chapter 2.

Whereas chapter 2 studies the behaviour of insects that are able to vary clutch sizes, egg sizes and the number of clutches, chapter 3 will study insects that cannot vary egg size or clutch size and that lay only one clutch, the size of which depends on the amount of competition experienced in the larval stage. The insects are also assumed to have discrete generations. An iteration model will be constructed to study the effects of density-dependent fecundity.

Chapter 4 will summarize and analyze an experiment conducted on *Pieris brassicae* to measure variation in egg size and density-related larval survival probability in this species.

Chapter 2

Maximization of expected total fitness gain and transformation of resources

2.1 Introduction

In section 1.4 of the literature review a model developed to optimize the total fitness gain of a female was discussed. We will now study this further.

Essentially three types of models were discussed so far: models that maximized either fitness per clutch or the rate at which fitness is gained or a female's expected total fitness gain.

The latter would appear to be the one that is closest to reality. It allows the inclusion of a variety of different factors which in combination would be expected to influence the oviposition behaviour exhibited by a female insect. In contrast to the optimization of the rate at which fitness is gained, where a constant or average clutch size is optimized, it allows variation in clutch size and egg size. Also, it seems more appropriate for species which lay fewer eggs. It can be used to model insects which emerge with a fixed amount of reproductive resources and which do not need to forage in order to mature eggs. It can also model the oviposition period in insects which forage and oviposit alternately.

2.2 Further discussion of the optimization of expected total fitness gain

Recall that if it is assumed that a female insect has accumulated a total reproductive reserve of M and if the probability of surviving to lay the i^{th} clutch is given by $p(i)$, her expected total fitness gain can be described by

$$w(n_1, ...n_k, m_1, ..., m_k) = \sum_{i=1}^{k} p(i) \cdot n_i s(n_i) \cdot f(m_i), \tag{2.1}$$

where k denotes the maximum possible number of clutches the female can produce given her initial resources M and where n_i and m_i are the clutch size and egg size for the i^{th} clutch. s and f are as described in chapter 1. The function w is to be optimized under the constraint $M = \sum_{i=1}^{k} n_i m_i$.

This model has been studied by Begon and Parker (1986). Using the Lagrange multiplier method and assuming an increasing mortality risk, they found that if clutch size is constrained, egg size should decline and vice versa. The model they discussed was, however, designed to find the optimal oviposition behaviour under the assumption that both clutch size and egg size vary. Constraining one of the variables in the equations derived from this model and solving for the other does not lead to an optimal solution. Instead one has to include the constraint into the model beforehand and then apply the Lagrange multiplier method with respect to the new constraint. This leads to different equations. The qualitative result however is the same as will be seen below. We will also show that the optimal oviposition behaviour in the case where both clutch size and egg size vary has the property that *both* clutch size and egg size decrease throughout the female's lifetime.

2.2.1 Optimal oviposition behaviour when egg size or clutch size are constrained

First set $n_1 = ... = n_k =: n$ in equation (2.1). Applying Lagrange's method of optimization subject to the constraint $M = \sum_{i=1}^{k} n_i m_i = n \sum_{i=1}^{k} m_i$ to this equation leads to

$$f'(m_i) = \frac{\lambda}{p(i)s(n)} \quad (i = 1, ..., k) \tag{2.2}$$

and

$$(ns'(n) + s(n)) = \lambda \frac{\sum_{j=1}^{k} m_j}{\sum_{j=1}^{k} p(j) f(m_j)}. \tag{2.3}$$

Now set $m_1 = ... = m_k =: m$ in equation (2.1). Applying Lagrange's method of optimization subject to the constraint $M = \sum_{i=1}^{k} n_i m_i = m \sum_{i=1}^{k} n_i$ to this equation leads to

$$f'(m) = \lambda \frac{\sum_{j=1}^{k} n_j}{\sum_{j=1}^{k} p(i) n_j s(n_j)} \tag{2.4}$$

and

$$(n_i s'(n_i) + s(n_i)) = \lambda \frac{m}{p(i) f(m)} \quad (i = 1, ..., k). \tag{2.5}$$

From equation (2.2) Begon and Parker proved that egg size decreases when clutch size is fixed. From equation (2.5) they proved that clutch size decreases when egg size is fixed for the most plausible forms of s. Thus, these results are still true if egg size or clutch size are set constant in the original model. But equation (2.3) and (2.4) differ from those obtained when Lagrange is applied to the model in which egg size or clutch size are not constrained. This means that the optimal oviposition behaviour will be different.

2.2.2 Optimal oviposition behaviour when egg size and clutch size vary

Applying Lagrange's method of optimization subject to the constraint $M = \sum_{i=1}^{k} n_i m_i$ to equation (2.1), leads to the following equations:

$$\begin{aligned} f'(m_i) s(n_i) &= \frac{\lambda}{p(i)} & (2.6) \\ \frac{f(m_i)}{m_i} (n_i s'(n_i) + s(n_i)) &= \frac{\lambda}{p(i)}, & (2.7) \end{aligned}$$

where $i = 1, ..., k$. For fixed i these two equations combine to

$$\frac{f(m_i)}{m_i} (n_i s'(n_i) + s(n_i)) = s(m_i) f'(n_i)$$

which is equivalent to

$$m_i \frac{f'(m_i)}{f(m_i)} - n_i \frac{s'(n_i)}{s(n_i)} = 1. \tag{2.8}$$

Note, that if the $n_i' s$ were kept constant in this equation, as Begon and Parker suggested, then either the $m_i' s$ would have to stay constant, too, or the term $x \frac{f'(x)}{f(x)}$ would have to be constant. Similarly, if one were to keep the $m_i' s$ constant, then either the $n_i's$ would also have to be constant or the term $x\frac{s'(x)}{s(x)}$ would have to stay constant.

It will be shown below that the two terms $x\frac{f'(x)}{f(x)}$ and $x\frac{s'(x)}{s(x)}$ are not constant, hence if egg sizes are kept constant the clutch sizes will be constant and vice versa. But the constant solutions, $n = n_1 = \cdots = n_k$ and $m = m_1 = \cdots = m_k$, would contradict equation (2.6) since the right hand side of it would vary whereas the left hand side would stay constant.

Proposition 2.2.1
Let $(n_1, ...n_k, m_1, ..., m_k)$ be a solution of the above optimization problem. Under the assumption that the female's survival probability decreases throughout her lifetime, i.e. under the assumption that $p(1) > p(2) > ... > p(k)$, both, egg size and clutch size, will decline as time (i.e. the sum index i) increases.

To prove this one needs the following lemma.

Lemma 2.2.1
Define the functions

$$F(m) := m\frac{f'(m)}{f(m)} \quad \text{and} \quad S(n) := -n\frac{s'(n)}{s(n)}.$$

Then for all possible functions s and f described in chapter 1, F will decrease monotonically and S will increase monotonically.

Proof:
It can be easily verified that for all s described in chapter 1, S' is a positive function and hence S increases monotonically. To show that F' is a negative function one needs to assume that $m > 1$. But since it does not matter in which units egg size is measured one can assume that $m_0 = 1$, and hence $m > 1$, without loss of generality.

□

Proof of the proposition:
Suppose that $(n_1, ...n_k, m_1, ..., m_k)$ is a solution of the optimization problem.

First look at equation (2.6). Since the female's survival probability decreases as i increases, the right hand side increases. That means that the left hand side must increase, too. Thus, at least one of the two factors on the left hand side has to increase. Now, from the shape of the two functions s and f (see chapter 1) one sees that $s(n_i)$ increases if and only if n_i decreases. Similarly, $f'(m_i)$ increases if and only if m_i decreases. We can therefore conclude that either the $m_i's$ or the $n_i's$ have to decrease.

Now, look at equation (2.8) which also has to be satisfied for $(n_1, ...n_k, m_1, ..., m_k)$ to be a solution. Using the definitions in the above lemma we can write it as

$$F(m_i) + S(n_i) = 1.$$

Since according to the lemma one of the terms on the left hand side is monotonically increasing whereas the other one is decreasing, it follows that either both, the $m_i's$ and the $n_i's$, increase or that both decrease. Together with what was concluded from equation (2.6), this proves that both decrease throughout the female's lifetime.

□

As Begon and Parker pointed out in their paper, equation (2.8) is the same as the one from which optimal egg size and clutch size are calculated when fitness is maximized per clutch (see chapter 1, section 1.2.3). This means that the optimal arrangement of egg size and clutch size is independent of adult mortality and so solving the above problem is equivalent to optimizing the allocation of resources $M_i := n_i \cdot m_i$ to the clutches that the female will lay during her life. Using this notation an equivalent way of defining fitness is

$$w(M_1, ..., M_k) = \sum_{i=1}^{k} p(i) \cdot n(M_i)s(n(M_i)) \cdot f(m(M_i)), \qquad (2.9)$$

where n and m are now functions which return the optimal clutch size and egg size given a particular amount of resources.

The above proposition can be restated as

Proposition 2.2.2
Let $(M_1, ..., M_k)$ be a solution of the optimization problem given by equation (2.9). Under the assumption that the female's survival probability decreases throughout her lifetime, i.e. under the assumption that $p(1) > p(2) > ... > p(k)$, the resources that are invested into the clutches will decline as time (i.e. the sum index i) increases. For each clutch egg size and clutch size are as if fitness were maximized per clutch.

Remark:

The summation index i is usually assumed to represent time. This, however, need not be so. The order in which the clutches occur in time is determined by the assumption that the adult survival probability decreases with time. If the weights $p(i)$ were to represent a different quantity, for example the quality of oviposition patches, and if this quantity increased with time then the model would predict that egg sizes and clutch sizes would increase, too. If the variation of the weights were more complicated than that, the model would predict that the variation in egg sizes and clutch sizes would parallel that of the weights.

2.3 A different approach to the problem

An organism's fitness is defined to be the ability to pass genes on to all future generations. According to this definition the number of offspring in *all* future generations should ideally be used as a measure for fitness. This is however impracticable because of its recursive definition. The number of children is not an accurate measure for fitness since it does not include the fitness of the children.

Recall that in all models mentioned so far the following quantity is used as a measure for clutch fitness:

$$\begin{pmatrix}\text{number of eggs}\\ \text{in the clutch}\end{pmatrix} \cdot \begin{pmatrix}\text{survival probability of}\\ \text{a larva of the clutch}\end{pmatrix} \cdot \begin{pmatrix}\text{fecundity of a larva}\\ \text{of the clutch}\end{pmatrix},$$

where the second and third factor are represented by the function s.

In the following I will discuss whether this quantity is a good approximation of fitness or whether it can be improved.

In the literature it is not clearly defined what the third factor represents nor how exactly it is to be measured. Parker and Begon (1986) refer to it as relative reproductive success as adult or as reproductive prospects. Ives (1989) defines it as fecundity of females or mating success of males - two quite different concepts. Parker and Courtney (1984) for heuristic purposes neglect this factor.

In general fecundity is measured in laboratory experiments by counting the number of offspring produced by the adults. This, however, might not reflect accurately the realized fecundity, i.e. the number of offspring that an adult female will be able to produce under natural circumstances. In general, a female needs energetic resources for a variety of different activities all of which are vital if she is to survive to reproduce: searching for food, catching prey, escaping from predators, surviving in adverse environmental conditions, maturing oocytes, mating, searching for oviposition sites, etc. Of all these activities, those that are directly related to reproduction form only a small part. An adult that produces offspring in a laboratory experiment might not be able to reproduce in the field.

Suppose that realized fecundity is included as the third factor in the above model. This means that the model essentially maximizes the number of grandchildren that a female can produce. First of all, it will in general be difficult, if not impossible, to measure realized fecundity. But there are also other reasons why the optimized quantity might still not be a suitable measure of an adult female's fitness:

- **Fecundity of males and females is not distinguished**
  It is not clear how the fecundity of males could be included into the model. On the other hand, not including male fecundity would essentially mean that males would have to be neglected when the fitness gain of a clutch is calculated even though they contribute to the competition among the larvae.

- **The strategy adopted by the offspring is not included**
  The fecundity of a larva that has reached adulthood is determined only from the number of offspring that it produces. A larva that produces few but fit offspring as adult would be considered to be less fecund than one that produces more but less fit offspring. On the other hand, the model might well predict that for the adult female it is optimal to produce relatively few but fit offspring. This means that the model contains two contradicting definitions of fecundity. It also means that it is assuming the very quantity that it is supposed to be determining.

- **The problem caused by the recursive definition of fitness is not resolved**
  The model maximizes the expected number of grandchildren. Maximizing the number of grandchildren that *will* be produced does not solve the problem caused by the iterative definition of fitness. If it is assumed that a female's offspring will adopt the same strategy as the female, i.e. optimize the number of grandchildren, it is not clear what exactly is optimized. For, even though the female optimizes the number of her grandchildren, her children optimize the number of their grandchildren, the female's great grandchildren. And hence the number of grandchildren for which the parent female's behaviour had been optimized, will, in general, not actually be produced. The assumption that a female's offspring will adopt the same strategy as the female could therefore lead to some kind of parent-offspring conflict (see Godfray (1995)).

I believe that instead of maximizing the number of grandchildren produced by a female one should maximize the expected total amount of energetic resources that the female's offspring will be able to acquire until they reach adulthood. The reasons for this are:

(1) Larvae that acquire more energetic resources through feeding will be expected to have a larger body size. Body size, on the other hand, has in many cases been shown to be positively correlated with the number of offspring that a female can produce and with her longevity. This would

suggest that in many cases the amount of energetic resources that an insect can gain during the larval stage is directly proportional to its realized fecundity as adult.

(2) The simplification made by neglecting the sex of the larvae seems unrealistic in the case where a larva's fecundity is included in the model. Male and female fecundity are two very different concepts and cannot be measured in the same units. In the case where the expected total resource gain is maximized the resources gained by male and female larvae can obviously be measured in the same units. The amount of energetic resources necessary for development and survival might, however, differ between the sexes. In a lot of insect species, for instance, females are larger than males and hence it would be expected that they need to acquire more resources than males. Such variation can be included in a model based on the optimization of the expected total amount of larval resource gain by including the sex ratio and by weighting male and female resource gains accordingly.

(3) The optimized quantity is independent of the strategy adopted by the offspring.

(4) If an adult female's oviposition behaviour is a response to factors in the environment a female might be able to assess the amount of food that the larvae will be able to acquire, for instance from the quality of the patch on which the eggs are oviposited. It has been shown that females do discriminate between oviposition sites of different quality (see for example Rothschild and Schoonhoven (1977)). The number of grandchildren, on the other hand, will in general be impossible to assess by an adult female.

The above suggests that the expected total amount of energetic resource gain obtained by a female's offspring might be a suitable measure for the fitness gain that she obtains by adopting a particular oviposition behaviour. There are, however, questions that need to be considered:

- A model derived from these ideas would be based on the assumption that the benefits obtained through optimizing resource gain are linearly related to the resource gain. This need not be so. The benefits from additional resource gain might be lower if the larva has already acquired a large amount of resources. Also, there might be costs, for example to large body size. There might be ways of incorporating a non-linear functional relationship between energetic resource gain and the resulting benefits into an optimization model. For simplicity, it will be assumed throughout the rest of this chapter that the relationship is linear.
- Optimizing energetic resource gain would not be suitable if it can occur that larvae gain enough resources to survive to adulthood but not enough to be physiologically capable of reproducing. This might be the case more

often in laboratory experiments. In the field one would expect that a larva that survived to adulthood should also be physiologically capable of reproducing.

- A trade-off between producing many weak and fewer but fitter larvae exists if, as would in general be expected, the per capita resource gain of the larvae of a clutch decreases with increasing clutch size. In this case, maximizing the total resource gain obtained from the clutch will result in an optimal clutch size that lies between the two extremes. However, it does not necessarily follow that the per capita resource gain of the larvae of that clutch will be large enough for them to survive and to reproduce. If the model includes a function that determines the per capita resource gain from the clutch size, say, this can be resolved by defining this function accordingly (for example such that the non-zero values it returns are larger than some minimum value).

A model that reflects the ideas presented in this section should result in an oviposition behaviour for which the resources, available to the female for reproduction, are transformed in an optimal way into resources gained by the female's offspring.

2.3.1 A simple model

The simplest way to derive a model which is based on the maximization of energetic resource gain is to extend the model given by equation (2.9). We will assume that egg size is fixed and neglect the factor $f(m(M_i))$. Now, let us further assume that resources can be invested in units, where one unit corresponds to one egg so that clutch size and invested resources become essentially the same concepts. As before it is assumed that the female has initially a fixed amount of resources and does not accumulate any more resources after the oviposition period has begun.

We will define a function T which returns the number of resource units a larva can gain before it reaches adulthood. T is assumed to depend on the clutch size, only.

Denote the total amount of resources gained by adopting a particular oviposition behaviour by $\tilde{R}$. Then $\tilde{R}$ is given by

$$\tilde{R}(M_1, ..., M_k) = \sum_{i=1}^{k} p(i) \cdot M_i s(M_i) \cdot T(M_i), \qquad (2.10)$$

where as before M_i is the amount of resources invested into the i^{th} clutch.

In the following section we will use the method of dynamic programming to solve this optimization problem for varying mortality risks and larval survival probabilities.

2.3.2 Simulation

The optimization problem given by equation (2.10) satisfies the conditions required for the dynamic programming method to be applicable, i.e. the state separation property and the separability of $\tilde{R}$. For further details concerning the dynamic programming process and its implementation please refer to the C++-program printed out in the appendix.

Equation (2.10) will be solved for the following cases:

- **Exponentially declining mortality risk**

  The probability that the female is alive to lay the i^{th} clutch is assumed to be given by

  $$p(i) := p^i,$$

  where $0 < p \leq 1$. The program will be run for $p = 0.99$, $p = 0.93$ and $p = 0.5$.

- **Varying competition among the larvae of a clutch**

  The survival probability of a larva from a clutch of size n will be described by the function

  $$s(n) := b \cdot n(1 - \frac{n}{v})^{\sigma}.$$

  For all cases we will set $b = 1$ and $v = 100$. This corresponds to a relatively high survival probability for a larva that hatched from a singly laid egg and to zero survival probability for a clutch of 100 eggs. The parameter σ determines the type and the severity of the competition. For large σ competition is high and the Lack clutch size is small. For small σ competition is low and the Lack clutch size is large. The program will be run for $\sigma = 3$, $\sigma = 1.5$, $\sigma = 1$, $\sigma = 0.66666$ and $\sigma = 0.33333$.

- **Different types of larval resource gain**

  The program will be run for seven different types of larval resource gain. T1 and T2 describe linear decline of larval resource gain. T3 and T4 describe exponential and logarithmic decline. T5 is motivated by Parker and Courtney's model of mortality anticipation (see the literature review for a brief description): larval resource gain is assumed to be fairly constant for clutch sizes below a certain threshold value; above the threshold value, which corresponds to the case where the food patch does not anymore provide sufficient food for all larvae, larval resource gain declines sharply. T6 and T7 describe Allee-effects. T6 is a parabola that interpolates the points $(1, \frac{3}{4}R_{max})$, $(\frac{v}{4}, R_{max})$ and $(v, 0)$, where $R_{max} = 100$. T7 is a Gaussian multiplied by a parabola that is designed such that T7 goes through the points $(1, \frac{3}{4}R_{max})$, $(\frac{v}{4}, R_{max})$ and $(v, 0)$, where $R_{max} = 100$. All curves are scaled such that the maximum resource gain is 100. See the tables for detailed definitions.

Figure 2.1 shows all seven functions. Figure 2.2 shows the fitness gained from a clutch, given these functions, i.e.

$$n \cdot s(n) \cdot T_j(n) \quad \text{, where } j = 1, ..., 7.$$

The parameters in s are set to $b = 1$, $v = 100$ and $\sigma = 1$ for these plots. The program is also run for the case in which the number of offspring is optimized. In that case the fitness gain from one clutch is given by the curve $n \cdot s(n)$ which corresponds to the dotted curve in figure 2.2.

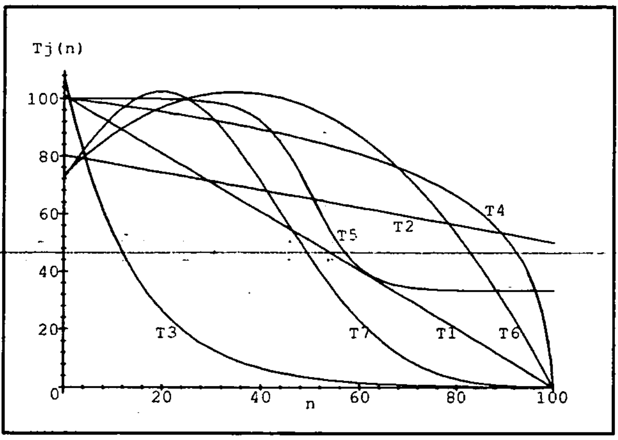


Figure 2.1: Functional representations of larval resource gain: T_j, where $j = 1, ..., 7$.

Apart from the clutch sizes all numbers generated by the program will be floating point numbers. There are several reasons for this. One is that the quantity that is to be optimized is an expected value and one cannot truncate or round terms of which this quantity is composed. Another reason is that even though floating point numbers seem unrealistic they represent the general trend, which is what we are interested in here, more accurately. Also, the continuous functions used to represent survival probability and resource gain, are built such that

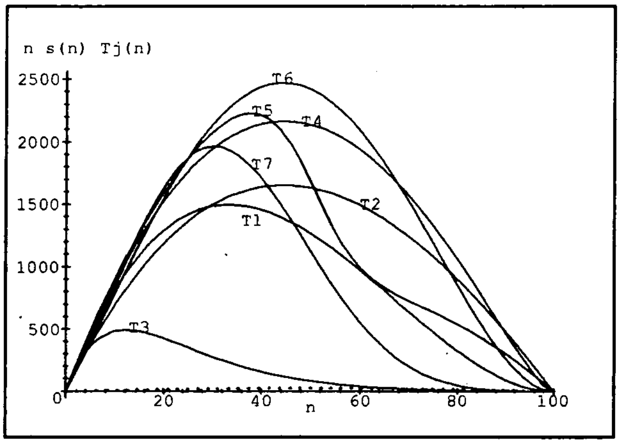


Figure 2.2: Fitness gained from a clutch of size n: $n \cdot s(n) \cdot T_j(n)$, where $j = 1, ..., 7$.

they describe reality qualitatively rather than quantitatively. And therefore, converting values returned by these functions to integer values would mean taking their use too far.

In the following tables L stands for the Lack clutch size. Underlined numbers serve merely as a check and denote the number of zero clutches the program found following the last non-zero clutch. Zero clutches followed by non-zero clutches would have been printed out singly. They do, however, not occur as would be expected from the fact that the female's mortality risk declines. For each of the seven functions that describe larval resource gain, the last table shows the total resource gain obtained by a female that optimizes the number of offspring.

p	σ	L	Strategies
0.99	0.33333	75	15 14 13 12 10 9 8 7 5 4 2 1 88
	0.66666	60	11 10 10 9 9 8 7 7 6 5 5 4 3 3 2 1 84
	1	50	9 9 8 8 7 7 7 6 6 5 5 4 4 3 3 3 2 2 1 1 80
	1.5	40	8 7 7 7 6 6 6 6 5 5 5 4 4 4 3 3 3 2 2 2 2 1 1 1 76
	3	25	5 5 5 5 5 5 5 4 4 4 4 4 4 4 3 3 3 3 3 3 2 2 2 2 2 2 1 1 1 1 1 1 1 67
0.93	0.33333	75	33 28 21 14 4 95
	0.66666	60	25 22 19 15 11 7 1 93
	1	50	21 19 17 14 12 9 6 2 92
	1.5	40	18 16 15 13 11 10 8 5 3 1 90
	3	25	13 12 11 10 10 9 8 7 6 5 4 3 2 87
0.5	0.33333	75	62 38 98
	0.66666	60	50 39 11 97
	1	50	43 36 21 97
	1.5	40	36 31 23 10 96
	3	25	24 23 21 17 12 3 94

Table 2.1: Optimization of the number of offspring. p: mortality risk, σ: larval competition, L: Lack clutch size.

p	σ	L	expected no of offspring	total resource gain	number of clutches	average clutch size	standard deviation
0.99	0.33333	75	91.8742	1	12	11.1111	2.06846
	0.66666	60	88.9904	1	16	10.4167	2.00693
	1	50	86.8368	1	20	10	2.26274
	1.5	40	84.2684	1	24	10.4167	2.59005
	3	25	78.7413	1	33	12.1212	3.32192
0.93	0.33333	75	76.8797	1	5	26.6667	3.05796
	0.66666	60	70.6575	1	7	23.8095	3.4858
	1	50	66.2312	1	8	25	3.47563
	1.5	40	61.1865	1	10	25	4.28661
	3	25	51.2236	1	13	30.7692	4.87884
0.5	0.33333	75	30.5545	1	2	66.6667	2.26274
	0.66666	60	24.0341	1	3	55.5556	4.73951
	1	50	20.0888	1	3	66.6667	3.1791
	1.5	40	16.1342	1	4	62.5	4.91172
	3	25	10.0929	1	6	66.6667	7.17031

Table 2.2: Optimization of the number of offspring. p: mortality risk, σ: larval competition, L: Lack clutch size.

p	σ	L	Strategies
0.99	0.33333	55	9 9 8 8 8 7 7 6 6 5 5 4 4 3 3 2 2 2 1 1 80
	0.66666	46	8 8 7 7 7 6 6 6 5 5 5 4 4 4 3 3 3 2 2 2 1 1 1 77
	1	39	7 7 7 6 6 6 6 5 5 5 5 4 4 4 3 3 3 3 2 2 2 2 1 1 1 75
	1.5	32	6 6 6 6 6 5 5 5 5 5 4 4 4 4 3 3 3 3 3 2 2 2 2 2 1 1 1 1 72
	3	22	5 5 5 5 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 1 1 1 1 1 1
			1 1 64
0.93	0.33333	55	22 19 17 14 12 9 5 2 92
	0.66666	46	19 17 15 14 12 9 7 5 2 91
	1	39	17 16 14 13 11 9 8 6 4 2 90
	1.5	32	15 14 13 12 11 9 8 7 5 4 2 89
	3	22	11 11 10 10 9 8 8 7 6 5 5 4 3 2 1 86
0.5	0.33333	55	45 36 19 97
	0.66666	46	39 34 23 4 96
	1	39	35 31 23 11 96
	1.5	32	30 27 23 16 4 95
	3	22	21 20 19 17 14 8 1 93

Table 2.3: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_1(x) := -\frac{100-33}{v}(x-1)+100$.

p	σ	L	expected no of offspring	total resource gain	number of clutches	average clutch size	standard deviation
0.99	0.33333	55	90.8021	8744.03	20	9.09091	2.13588
	0.66666	46	88.3759	8564.67	23	9.4518	2.31312
	1	39	86.4465	8409.74	25	10.2564	2.48599
	1.5	32	84.0453	8208.36	28	11.1607	2.81002
	3	22	78.6615	7735	36	12.6263	3.5855
0.93	0.33333	55	74.674	6687.49	8	22.7273	3.32285
	0.66666	46	69.5154	6327	9	24.1546	3.52475
	1	39	65.5327	6028.12	10	25.641	3.90552
	1.5	32	60.8292	5654.25	11	28.4091	4.24942
	3	22	51.0369	4840.84	15	30.303	5.55464
0.5	0.33333	55	28.4046	2080.83	3	60.6061	3.39502
	0.66666	46	23.1276	1771.7	4	54.3478	5.84132
	1	39	19.5481	1547.4	4	64.1026	4.70008
	1.5	32	15.825	1299.03	5	62.5	6.48014
	3	22	9.9898	871.127	7	64.9351	8.1746

Table 2.4: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_1(x) := -\frac{100-33}{v}(x-1)+100$.

p	σ	L	Strategies
0.99	0.33333	67	11 10 10 9 8 8 7 7 6 5 5 4 3 3 2 1 1 83
	0.66666	53	9 9 8 8 7 7 7 6 6 5 5 4 4 3 3 3 2 2 1 1 80
	1	44	8 8 7 7 7 6 6 6 5 5 5 4 4 4 3 3 3 2 2 2 1 1 1 77
	1.5	36	7 7 6 6 6 6 5 5 5 5 4 4 4 4 4 3 3 3 3 2 2 2 1 1 1 1 74
	3	23	5 5 5 5 5 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 1 1 1 1 1 1
			66
0.93	0.33333	67	26 22 19 15 11 6 1 93
	0.66666	53	21 19 17 14 12 9 6 2 92
	1	44	19 17 15 14 12 9 7 5 2 91
	1.5	36	16 15 14 12 11 9 8 6 5 3 1 89
	3	23	12 11 11 10 9 9 8 7 6 5 4 3 3 2 86
0.5	0.33333	67	54 39 7 97
	0.66666	53	44 36 20 97
	1	44	39 33 23 5 96
	1.5	36	33 29 24 14 96
	3	23	22 22 20 17 13 6 94

Table 2.5: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_2(x) := -\frac{80-50}{v}(x-1)+80$.

p	σ	L	expected no of offspring	total resource gain	number of clutches	average clutch size	standard deviation
0.99	0.33333	67	91.378	7126.86	17	8.77963	1.92164
	0.66666	53	88.7633	6957.05	20	9.43396	2.13466
	1	44	86.7018	6816.59	23	9.88142	2.41827
	1.5	36	84.1981	6638.81	26	10.6838	2.62621
	3	23	78.7173	6235.41	34	12.7877	3.36451
0.93	0.33333	67	76.025	5653.98	7	21.322	3.25438
	0.66666	53	70.2199	5298.29	8	23.5849	3.2789
	1	44	66.0189	5017.3	9	25.2525	3.68497
	1.5	36	61.0413	4678.89	11	25.2525	4.38246
	3	23	51.163	3968.39	14	31.0559	5.24836
0.5	0.33333	67	29.9656	1965.73	3	49.7512	5.06731
	0.66666	53	23.7852	1624.76	3	62.8931	3.26075
	1	44	19.9331	1388.96	4	56.8182	5.85641
	1.5	36	16.0718	1144.59	4	69.4444	3.94796
	3	23	10.063	743.807	6	72.4638	6.0766

Table 2.6: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_2(x) := -\frac{80-50}{v}(x-1)+80$.

p	σ	L	Strategies
0.99	0.33333	14	4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 51
	0.66666	13	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 51
	1	12	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 50
	1.5	12	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 49
	3	10	3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 45
0.93	0.33333	14	8 8 8 7 7 7 6 6 6 5 5 5 4 4 3 3 2 2 2 1 1 79
	0.66666	13	8 8 7 7 7 7 6 6 6 5 5 5 4 4 3 3 3 2 2 1 1 79
	1	12	8 8 7 7 7 6 6 6 6 5 5 5 4 4 3 3 3 2 2 1 1 1 78
	1.5	12	8 7 7 7 7 6 6 6 5 5 5 5 4 4 4 3 3 2 2 2 1 1 78
	3	10	7 7 6 6 6 6 6 5 5 5 5 4 4 4 4 3 3 3 2 2 2 2 1 1 1 75
0.5	0.33333	14	13 13 13 13 13 12 10 8 5 91
	0.66666	13	13 13 13 12 12 11 10 8 6 2 90
	1	12	12 12 12 12 12 11 10 9 7 3 90
	1.5	12	11 11 11 11 11 11 10 9 8 5 2 89
	3	10	10 10 10 10 10 9 9 9 8 7 5 3 88

Table 2.7: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_3(x) := 100e^{-0.07(x-1)}$.

p	σ	L	expected no of offspring	total resource gain	number of clutches	average clutch size	standard deviation
0.99	0.33333	14	82.2863	7423.32	49	14.5773	4.28085
	0.66666	13	81.3492	7362.93	49	15.6986	4.34587
	1	12	80.4793	7304.03	50	16.6667	4.71405
	1.5	12	79.3077	7217.88	51	16.3399	4.70826
	3	10	75.7929	6975.85	55	18.1818	5.4938
0.93	0.33333	14	58.0044	3980.52	21	34.0136	7.485
	0.66666	13	56.2813	3895.71	21	36.63	7.68498
	1	12	54.7008	3814.3	22	37.8788	8.71838
	1.5	12	52.3772	3699.26	22	37.8788	8.05599
	3	10	46.1121	3395.3	25	40	9.38083
0.5	0.33333	14	12.3176	533.948	9	79.3651	5.84183
	0.66666	13	11.6742	509.924	10	76.9231	8.4265
	1	12	10.5008	487.913	10	83.3333	7.45356
	1.5	12	9.20804	458.044	11	75.7576	7.94552
	3	10	7.26806	388.042	12	83.3333	7.52773

Table 2.8: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_3(x) := 100e^{-0.07(x-1)}$.

p	σ	L	Strategies
0.99	0.33333	62	12 11 11 10 9 8 8 7 6 5 4 3 3 2 1 85
	0.66666	52	10 9 9 8 8 7 7 6 6 5 5 4 4 3 3 2 2 1 1 81
	1	45	8 8 8 7 7 7 6 6 6 5 5 4 4 4 3 3 3 2 2 1 1 79
	1.5	37	7 7 7 6 6 6 6 5 5 5 5 4 4 4 3 3 3 3 2 2 2 2 1 1 1 75
	3	24	5 5 5 5 5 5 4 4 4 4 4 4 4 3 3 3 3 3 3 3 2 2 2 2 2 2 2 1 1 1 1 1 1 1
			66
0.93	0.33333	62	27 24 20 15 10 4 94
	0.66666	52	22 20 17 15 12 8 5 1 92
	1	45	19 18 16 14 12 9 7 4 1 91
	1.5	37	17 15 14 13 11 10 8 6 4 2 90
	3	24	12 12 11 10 10 9 8 7 6 5 4 3 2 1 86
0.5	0.33333	62	53 40 7 97
	0.66666	52	44 36 20 97
	1	45	39 34 23 4 96
	1.5	37	33 30 24 13 96
	3	24	23 22 20 17 13 5 94

Table 2.9: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_4(x) := \frac{100}{\ln(100)} \ln(-(x-101))$.

p	σ	L	expected no of offspring	total resource gain	number of clutches	average clutch size	standard deviation
0.99	0.33333	62	91.6409	9007.69	15	10.7527	2.14785
	0.66666	52	88.8835	8773.03	19	10.1215	2.30516
	1	45	86.786	8583.91	21	10.582	2.28586
	1.5	37	84.2476	8349.79	25	10.8108	2.62037
	3	24	78.7331	7827.13	34	12.2549	3.382
0.93	0.33333	62	76.3852	7264.59	6	26.8817	3.14137
	0.66666	52	70.4147	6776.13	8	24.0385	3.75862
	1	45	66.1221	6398.63	9	24.6914	3.95586
	1.5	37	61.1483	5949.59	10	27.027	4.00876
	3	24	51.2188	5024.39	14	29.7619	5.58573
0.5	0.33333	62	29.8922	2569.16	3	53.7634	5.40904
	0.66666	52	23.7852	2123.69	3	64.1026	3.32346
	1	45	19.9587	1811.51	4	55.5556	5.97112
	1.5	37	16.0884	1487.28	4	67.5676	4.13434
	3	24	10.0822	957.574	6	69.4444	6.3099

Table 2.10: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_4(x) := \frac{100}{\ln(100)} \ln(-(x-101))$.

p	σ	L	Strategies
0.99	0.33333	41	15 14 13 12 10 9 8 7 5 4 2 1 88
	0.66666	40	11 10 10 9 9 8 7 7 6 5 5 4 3 3 2 1 84
	1	38	9 9 8 8 7 7 7 6 6 5 5 4 4 3 3 3 2 2 1 1 80
	1.5	35	8 7 7 7 6 6 6 6 5 5 5 4 4 4 3 3 3 2 2 2 2 1 1 1 76
	3	25	5 5 5 5 5 5 5 4 4 4 4 4 4 4 3 3 3 3 3 3 2 2 2 2 2 2 1 1 1 1 1 1 1 67
0.93	0.33333	41	29 26 22 16 7 95
	0.66666	40	24 22 19 15 11 7 2 93
	1	38	21 19 17 14 12 9 6 2 92
	1.5	35	18 16 15 13 11 9 8 6 3 1 90
	3	25	13 12 11 10 10 9 8 7 6 5 4 3 2 87
0.5	0.33333	41	39 37 24 97
	0.66666	40	38 35 27 97
	1	38	36 33 26 5 96
	1.5	35	33 30 25 12 96
	3	25	24 23 21 17 12 3 94

Table 2.11: Optimization of total resource gain. . p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_5(x) := -\frac{b-100}{\tanh(\beta(1-a))}\tanh(\beta(x-a))+b$, where $\beta = 0.1$, $a = 50$ and $b = 66.67$

p	σ	L	expected no of offspring	total resource gain	number of clutches	average clutch size	standard deviation
0.99	0.33333	41	91.8742	9184.79	12	20.3252	3.78376
	0.66666	40	88.9904	8897.95	16	15.625	3.0104
	1	38	86.8368	8682.99	20	13.1579	2.97729
	1.5	35	84.2684	8426.37	24	11.9048	2.96005
	3	25	78.7413	7873.9	33	12.1212	3.32192
0.93	0.33333	41	76.7361	7634.23	5	48.7805	4.26655
	0.66666	40	70.6428	7051.13	7	35.7143	4.94614
	1	38	66.2312	6616.61	8	32.8947	4.5732
	1.5	35	61.1864	6115.43	10	28.5714	4.83187
	3	25	51.2236	5121.35	13	30.7692	4.87884
0.5	0.33333	41	27.2053	2573.1	3	81.3008	2.80929
	0.66666	40	23.117	2212.6	3	83.3333	2.01039
	1	38	19.7494	1917.77	4	65.7895	6.37038
	1.5	35	16.0903	1583.41	4	71.4286	4.58925
	3	25	10.0929	1006.29	6	66.6667	7.17031

Table 2.12: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_5(x) := -\frac{b-100}{\tanh(\beta(1-a))}\tanh(\beta(x-a))+b$, where $\beta = 0.1$, $a = 50$ and $b = 66.67$.

p	σ	L	Strategies
0.99	0.33333	55	26 25 25 24 96
	0.66666	49	21 21 20 19 19 95
	1	44	16 16 15 14 14 13 12 93
	1.5	38	13 12 12 11 11 10 9 8 8 6 90
	3	27	7 7 7 7 7 6 6 6 5 5 5 5 4 4 4 3 3 3 2 2 1 1 78
0.93	0.33333	55	36 33 31 97
	0.66666	49	29 27 24 20 96
	1	44	25 23 21 18 13 95
	1.5	38	21 20 18 16 14 11 94
	3	27	15 14 13 12 11 10 9 7 6 3 90
0.5	0.33333	55	50 44 6 97
	0.66666	49	44 37 19 97
	1	44	40 35 25 97
	1.5	38	35 32 26 7 96
	3	27	26 24 22 18 10 95

Table 2.13: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_6(x) := -\frac{19}{792}x^2 + \frac{1319}{792}x + \frac{14525}{198}$.

p	σ	L	expected no of offspring	total resource gain	number of clutches	average clutch size	standard deviation
0.99	0.33333	55	88.6223	8862.21	4	45.4545	0.25713
	0.66666	49	83.6394	8120.01	5	40.8163	0.408163
	1	44	82.3557	7605.17	7	32.4675	0.832841
	1.5	38	80.6487	7099.15	10	26.3158	1.74559
	3	27	77.4917	6321.08	22	16.835	3.38346
0.93	0.33333	55	75.8613	7747.55	3	60.6061	0.647096
	0.66666	49	69.3672	6938.66	4	51.0204	1.38415
	1	44	65.0086	6334.97	5	45.4545	2.13201
	1.5	38	59.7993	5679.17	6	43.8596	2.22261
	3	27	50.5334	4507.15	10	37.037	4.22287
0.5	0.33333	55	29.6441	2886.89	3	60.6061	6.1353
	0.66666	49	23.8084	2390.68	3	68.0272	3.72228
	1	44	20.0312	2035.07	3	75.7576	2.45483
	1.5	38	16.118	1636.46	4	65.7895	5.72935
	3	27	10.0549	998.847	5	74.0741	4.68486

Table 2.14: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_6(x) := -\frac{19}{792}x^2 + \frac{1319}{792}x + \frac{14525}{198}$.

p	σ	L	Strategies
0.99	0.33333	34	20 20 20 20 20 95
	0.66666	32	17 17 17 17 16 16 94
	1	30	15 15 15 14 14 14 13 93
	1.5	28	14 13 13 13 12 12 12 11 92
	3	23	8 8 8 8 8 7 7 7 6 6 6 5 5 4 4 3 84
0.93	0.33333	34	23 22 20 19 16 95
	0.66666	32	22 21 20 19 18 95
	1	30	20 19 18 16 15 12 94
	1.5	28	19 18 17 17 15 14 94
	3	23	15 14 13 12 12 11 9 8 6 91
0.5	0.33333	34	32 30 26 12 96
	0.66666	32	30 29 25 16 96
	1	30	29 28 25 18 96
	1.5	28	27 26 25 22 96
	3	23	22 22 21 19 15 1 94

Table 2.15: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_7(x) := (-0.00513202316x^2 - 0.6441830438x + 119.5503275)e^{-\frac{1}{12.5\cdot v}(x-\frac{v}{4})^2}$.

p	σ	L	expected no of offspring	total resource gain	number of clutches	average clutch size	standard deviation
0.99	0.33333	34	90.0837	9223.94	5	58.8235	0
	0.66666	32	85.5271	8678.97	6	52.0833	0.360844
	1	30	82.3827	8208.59	7	47.619	0.617213
	1.5	28	78.2653	7643.52	8	44.6429	0.874818
	3	23	75.5025	6626.38	16	27.1739	2.78397
0.93	0.33333	34	75.8062	7722.25	5	58.8235	1.61095
	0.66666	32	70.0428	7158.64	5	62.5	0.988212
	1	30	65.8192	6666.06	6	55.5556	2.19427
	1.5	28	59.7991	6065.02	6	59.5238	1.4869
	3	23	50.1383	4842.11	9	48.3092	3.60867
0.5	0.33333	34	24.3875	2244.67	4	73.5294	4.59426
	0.66666	32	21.0656	2000.38	4	78.125	3.45168
	1	30	18.6013	1792.99	4	83.3333	2.86744
	1.5	28	15.5348	1532.87	4	89.2857	1.33631
	3	23	10.0585	1025.41	6	72.4638	7.89024

Table 2.16: Optimization of total resource gain. p: mortality risk, σ: larval competition, L: Lack clutch size. Larval resource gain given by $T_7(x) := (-0.00513202316x^2 - 0.6441830438x + 119.5503275)e^{-\frac{1}{12.5\cdot v}(x-\frac{v}{4})^2}$.

p	σ	L	T1	T2	T3	T4	T5	T6	T7
0.99	0.33333	75	8578.37	7077.23	4783.89	8980.22	9184.79	8106.49	8673.65
	0.66666	60	8492.22	6937.07	5637.43	8763.18	8897.95	7535.43	7981.04
	1	50	8370.01	6806.5	6047.47	8579.76	8682.99	7193.31	7554.02
	1.5	40	8181.84	6631.77	6291.06	8346.16	8426.37	6861.55	7151.72
	3	25	7726.35	6233.13	6509.47	7825.9	7873.9	6241.38	6422.4
0.93	0.33333	75	6405.22	5576.01	1622.26	7207.15	7609.23	7601.54	7261.1
	0.66666	60	6202.26	5265.96	2177.88	6756.64	7050.83	6755.93	7035.84
	1	50	5954.4	4999.07	2475.11	6388.59	6616.61	6176.28	6568.98
	1.5	40	5607.53	4666.06	2693.16	5942.23	6115.43	5562.85	5960.57
	3	25	4824.75	3964.63	2865.41	5022.01	5121.35	4452.9	4755.69
0.5	0.33333	75	1925.65	1938.48	92.1658	2515.07	1638.21	2721.32	1055.15
	0.66666	60	1692.19	1604.27	163.237	2097.43	1831.86	2349.2	1340.07
	1	50	1496.03	1377.47	165.63	1799.99	1824.9	2024.53	1450.76
	1.5	40	1272.76	1138.2	203.989	1481.23	1571.18	1633.87	1400.58
	3	25	862.15	741.547	228.861	955.99	1006.29	994.382	1019.92

Table 2.17: Total resource gain of a female that optimizes the number of offspring. p: mortality risk, σ: larval competition, L: Lack clutch size.

2.3.3 Results

First note that in this simple model there is no benefit to laying clutches later in life as for example in Mangel's models (see Mangel (1987)) which include a trade-off between laying eggs while alive, i.e. as early as possible, and waiting until a more profitable host type has been encountered. This model studies the trade-off between laying eggs while alive and not reducing the offspring's survival prospects by laying too many eggs per clutch. Thus, as expected, clutch sizes decline.

- **Clutch sizes are substantially smaller than the Lack clutch size:** The simulation confirms that it is in general optimal for a female to produce clutches smaller than the Lack clutch size. In the cases that were studied here, the average clutch size is often 90% lower than the Lack clutch size. This occurs when the adult female's mortality is low and hence there is a low cost to laying eggs later in life. For T6 and T7, i.e. in the case of an Allee-effect, the average clutch size is substantially closer to the Lack clutch size. This can be explained by the fact that larval survival chances reach their maximum not when eggs are laid singly but for a larger clutch size that is closer to the Lack clutch size. In the case of an Allee-effect there is also less variation in clutch size.

- **Influence of adult mortality on oviposition behaviour:**
  For higher adult mortality, the expected total resource gain and the number of survivors are lower. It is optimal to lay fewer larger clutches.
- **Influence of larval competition on oviposition behaviour:**
  For higher competition among larvae, the expected total resource gain and the number of survivors are lower. There does not appear to be a pattern to the variation of average clutch size. The maximum clutch size, however, decreases as larval competition increases.
- **Allee-effect:**
  In the case of the two functions T6 and T7 that represent Allee-effects clutch sizes are closer to the Lack clutch size compared to decreasing survival probability functions. There is also considerably less variation of clutch size.

Optimization of expected total resource gain versus optimization of the number of offspring

The number of offspring produced by a female that maximizes the number of her offspring is never smaller than the number of offspring produced by a female that optimizes resource gain. Similarly, the expected total resource gain of females that maximize resource gain is never smaller than that of a female that maximizes the number of offspring. The simulation tables show that there is little difference between the number of offspring produced by two females, one of which maximizes the number of her offspring and one of which maximizes resource gain. One question is therefore whether the additional resource gain of a female that optimizes resource gain is comparable to the sacrifice constituted by the reduced number of offspring. To answer this question one has to somehow convert resource gain into offspring. This will be done by dividing the additional resource gain by the average per capita resource gain of the offspring. Subtracting from this the sacrificed offspring and transforming the resulting value into percentages of the maximum number of offspring that a female can produce, gives a measure for the costs or benefits obtained from maximizing resource gain rather than offspring:

$$C_j := \left(\frac{R_j - R_{j,0}}{\frac{R_j}{S_j}} - (S_0 - S_j) \right) \cdot \frac{100}{S_0}, \qquad (2.11)$$

where

- R_j denotes the resource gain (given by T_j) of a female that maximizes resource gain,
- $R_{j,0}$ denotes the resource gain (given by T_j) of a female that maximizes the number of offspring,
- S_j denotes the number of offspring produced by a female that maximizes resource gain (given by T_j),
- S_0 denotes the number of offspring produced by a female that maximizes the number of offspring.

Note, that the average per capita resource gain $\frac{R_j}{S_j}$ of the offspring produced by a female that maximizes the total resource gain is always larger than or equal to $\frac{R_{j,0}}{S_0}$, the per capita resource gain of the offspring produced by a female that maximizes the number of her offspring. Thus this measure represents an upper bound for the costs (or equivalently a lower bound for the benefits) that are to be expected when total resource gain is maximized.

For each value of p and σ and for each T_j the above quantity can be calculated from the data given in the simulation tables. The resulting values are shown in figures 2.3 to 2.9. In the diagrams the order (with respect to p and σ) of the values from left to right is the same as the order in the corresponding table from top to bottom.

The diagrams show that with respect to the measure defined in (2.11) there is generally no or very little cost to maximizing the expected total resource gain whereas substantial benefits do occur.

The costs for T1, T2, T4 and T5 are below 0.2% of S_0. The costs for T6 and T7 are below 1.5%. Only for T3 and only for the highest level of adult mortality are there substantial costs: for the lowest level of larval competition about 25%.

The benefits for T1, T2, T4 and T6 are below 3%. For T3, T5 and T7 the benefits can be as high as 20%. For T5 and T7 benefits are high when adult mortality is high, i.e. when clutch sizes are larger.

The curves $ns(n)T_5(n)$ and $ns(n)T_7(n)$ decline sharply when the Lack clutch size (with respect to $ns(n)T_j(n)$) is exceeded. But in the simulation Lack clutch sizes for resource maximization are smaller than the Lack clutch sizes for the maximization of the number of offspring. This means that for high adult mortality a female that optimizes the number of offspring would be expected to lay larger clutches which would result in a very low resource gain. The curve $ns(n)T_3(n)$ declines less sharply than all other curves which might explain why in this case there are high benefits to maximizing the number of offspring when adult mortality is high: larger clutches which are favoured when the number of offspring is optimized will result in a resource gain that is only slightly reduced compared to that gained when resource gain is maximized.

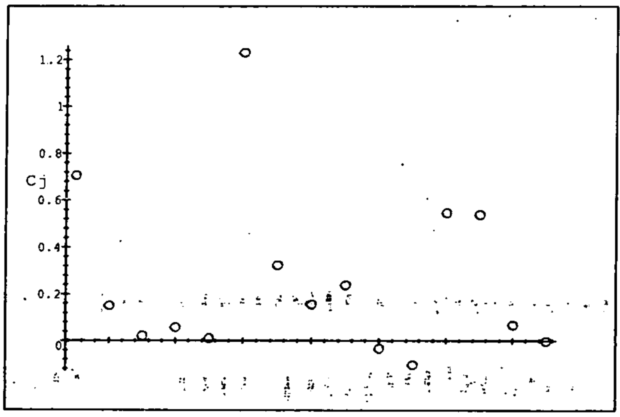


Figure 2.3: The data for this diagram were taken from tables 2.4, 2.2 and 2.17. The diagram shows the value of C_1 for each combination of p and σ. The order of the values from left to right is the same as the order in the corresponding tables from top to bottom.

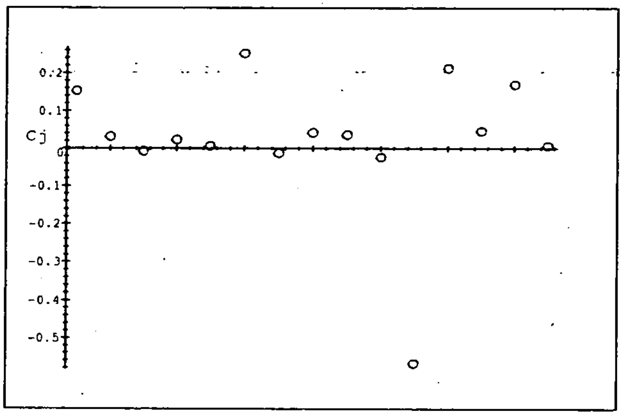


Figure 2.4: The data for this diagram were taken from tables 2.6, 2.2 and 2.17. The diagram shows the value of C_2 for each combination of p and σ. The order of the values from left to right is the same as the order in the corresponding tables from top to bottom.

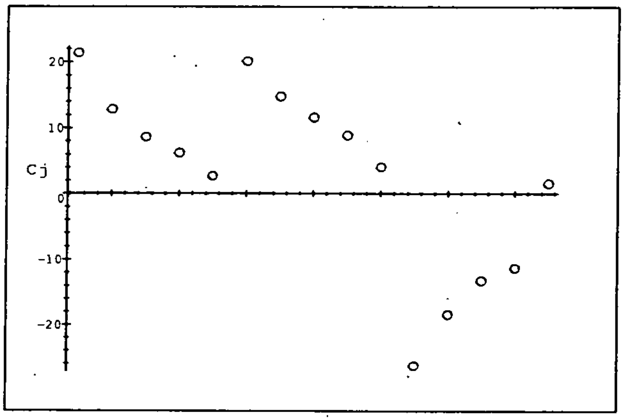


Figure 2.5: The data for this diagram were taken from tables 2.8, 2.2 and 2.17. The diagram shows the value of C_3 for each combination of p and σ. The order of the values from left to right is the same as the order in the corresponding tables from top to bottom.

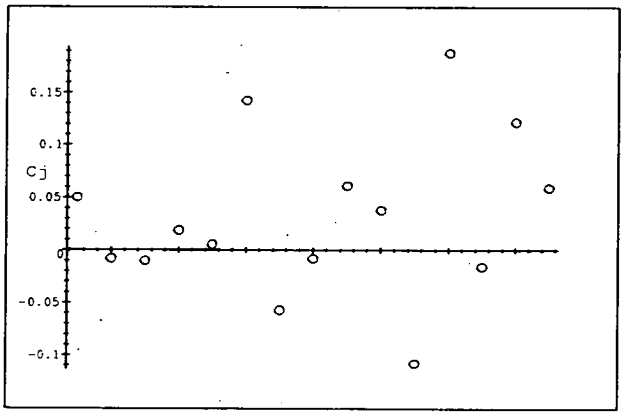


Figure 2.6: The data for this diagram were taken from tables 2.10, 2.2 and 2.17. The diagram shows the value of C_4 for each combination of p and σ. The order of the values from left to right is the same as the order in the corresponding tables from top to bottom.

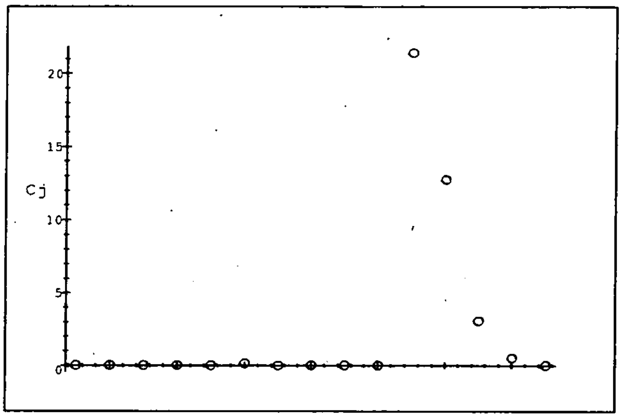


Figure 2.7: The data for this diagram were taken from tables 2.12, 2.2 and 2.17. The diagram shows the value of C_5 for each combination of p and σ. The order of the values from left to right is the same as the order in the corresponding tables from top to bottom.

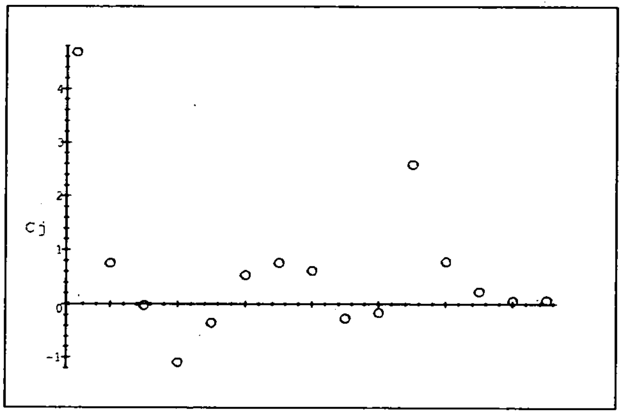


Figure 2.8: The data for this diagram were taken from tables 2.14, 2.2 and 2.17. The diagram shows the value of C_6 for each combination of p and σ. The order of the values from left to right is the same as the order in the corresponding tables from top to bottom.

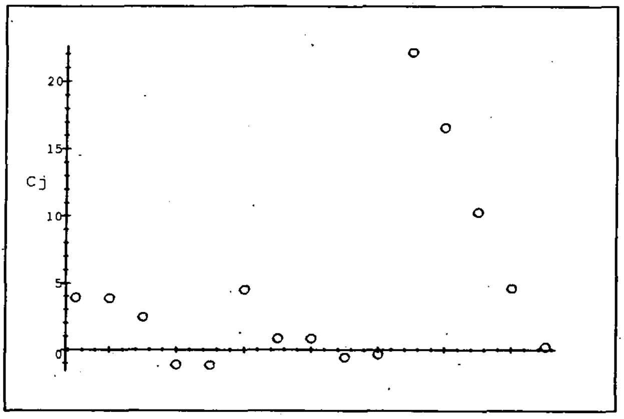


Figure 2.9: The data for this diagram were taken from tables 2.16, 2.2 and 2.17. The diagram shows the value of C_7 for each combination of p and σ. The order of the values from left to right is the same as the order in the corresponding tables from top to bottom.

2.4 Discussion

A further discussion of the model given by equation (2.1) showed that, assuming an increasing mortality risk, a female should invest decreasing amounts of reproductive resources with increasing age.This holds in the case where the female can vary both, the sizes of the clutches she lays and the sizes of her eggs, as well as in the case where she can vary only one of the two variables. In the latter case this was found by Begon and Parker (1986) but the proof was incorrect in that the constraint of keeping one of the two variables fixed was not incorporated into the model at the outset but into the equations derived from the model in which the variables were not constrained.

In the second part of this chapter a different measure for a female's fitness was discussed, the main idea being that it should be optimal for a female to maximize the number of her offspring but at the same time to ensure that her offspring will be as successful as possible. It was suggested that the success of the offspring should be expressed in terms of the amount of energetic resources that they can acquire as larvae because this would: • reduce the recursive definition of fitness to a non-recursive measure for fitness, • eliminate the conflict occurring between parent and offspring when the number of grandchildren is optimized, • allow the inclusion of realized fecundity rather than the fecundity measured in laboratory experiments, • be independent of the offspring's oviposition behaviour and • allow for the larvae's sex to be included.

It will in general be difficult to test whether a behaviour predicted to be optimal by an optimization model is optimal in reality (i.e. 'selected for'), in particular if in natural circumstances the variables that are to be optimized are related to factors that vary unpredictably as for example the quality of oviposition sites.

I believe that a model that is based on the maximization of the number of grandchildren, for the reasons listed in section 2.3, will contain logical flaws which would be eliminated in a model based on the maximization of larval energetic resource gain. To test whether a fitness measure based on larval resource gain would predict behaviours that represent a good approximation to oviposition behaviours occurring in nature one would have to determine the benefits that are associated with energetic resource gain and in particular what kind of functional relationship there is between larval resource gain and realized fecundity.

The simulation showed that for the simple model given by equation (2.1) there are no or very little costs but high benefits in some cases to optimizing the expected total amount of energetic resource gain that can be acquired by a female's offspring versus optimizing the expected total number of offspring. The simulation also confirmed that clutches are in general smaller than the Lack solution clutch size. In this particular case, moreover, they are often as much as 90% smaller. Data were generated for different types of resource functions to study effects of different phenomena such as Allee-effect or mortality anticipation on oviposition behaviour. The resulting oviposition behaviours do differ. It is, however, not clear how significant these differences really are considering the fact that the curves $n \cdot s(n) \cdot T_j(n)$ are qualitatively very similar in the interval $(1, n_j^*)$, where n_j^* is the Lack clutch size with respect to Tj.

Chapter 3

Effects of density-dependent fecundity on population size

3.1 Introduction

Previous work on the oviposition behaviour of insects suggests that a poor larval diet can result in lower adult fecundity and hence cause density-dependent population effects: larvae reared in a high density environment would later produce relatively few offspring. This in turn would result in the next generation of larvae growing up in a low density environment which would lead to adults with higher fecundity. For a species with almost non-overlapping generations this would suggest a cyclic population fluctuation.

In this chapter a mathematical model which reflects density-dependent adult fecundity will be developed. The resulting iteration will be studied for the possible occurrences of cycles or of convergence. In section 3.4 a small simulation will be run for different types and different levels of larval competition.

3.2 The model

Assumptions:

The following model is designed for a population that lives in an environment of constant carrying capacity. Seasonal changes, interaction with other species (e.g. predation) and other external factors will be neglected.

It is assumed that females lay only one clutch of eggs and that they do not influence the living conditions of their offspring. This means either that the adults do not affect the resources needed by the larvae or that the adults die before their offspring hatches. Furthermore adults from different generations are assumed to oviposit in discrete time intervals. These two assumptions imply that the generations do not overlap.

Finally the clutch sizes produced by females will be assumed to be constant within a generation.

These assumptions are made to keep the model mathematically simple enough to derive analytical results. The model is not designed to describe only one particular species but to study and explain general trends in an insect population.

Derivation of the model:

Note: In order to simplify the discussion of the model, variables will be treated as continuous variables even though in reality only integer values make sense.

Denote by F_g, n_g and N_g the number of females, their clutch size and the total number of eggs produced by these females in generation g. Suppose that N_g is given. An iteration will be obtained by calculating F_g and n_g from N_g. The total number of eggs in the following generation is then given by $N_{g+1} := F_g \cdot n_g$.

The function

$$s(x) := \begin{cases} b(1-\frac{x}{v})^r & ,0 \leq x \leq v \\ 0 & ,x > v, \end{cases}$$

where $0 < b \leq 1$ and $r > 0$, will be used to represent the survival probability of the larvae. s has been shown to be a relatively realistic description of larval competition (see Suzuki and Iwasa (1980), Ives (1991) and Parker and Courtney (1984)). Here, v determines which population densities can be supported by the environment: if the total number of eggs increases beyond v, there will be no survivors. Multiplying the above expression by N_g to obtain the number of surviving adults and then multiplying by α, the proportion of surviving adult females, gives F_g, the number of females in generation g:

$$F_g := \begin{cases} b\alpha N_g(1-\frac{N_g}{v})^r & ,0 \leq N_g \leq v \\ 0 & ,N_g > v. \end{cases}$$

Now assume that the relationship between N_g and n_g is described by a function h. h should be positive on $[0, v)$ and 0 otherwise. It also should have only finite values. Since we want the model to reflect density-dependent adult fecundity it makes sense to assume that h decreases strictly monotonically and also that h is bounded from above by some constant such that the clutch size assigned by the function to a female can never exceed the maximum clutch size which can be produced by an insect of the species that is being modeled. This will be achieved by setting $h(0) = M$, where M is a suitable positive constant. Finally, it will be assumed that h is continuous and differentiable.

Let N_0 be the starting point of the iteration. Then the iteration can be summarized by the following three equations:

$$\begin{aligned} F_g &:= b\alpha N_g(1 - \frac{N_g}{v})^r, && (3.1) \\ n_g &:= h(N_g), && (3.2) \\ N_{g+1} &:= F_g \cdot n_g, && (3.3) \end{aligned}$$

for $0 \leq N_g \leq v$.

Substituting the first two expressions into the third gives a fixed point iteration for N_g:

$$N_{g+1} := f(N_g), \tag{3.4}$$

with iteration function

$$f(x) := b\alpha x\left(1 - \frac{x}{v}\right)^r \cdot h(x), \tag{3.5}$$

for $0 \leq N_g \leq v$ and starting point N_0.

In this model α, v, and M are positive constants which are determined by the behaviour and the phenotype of the species under study. $r > 0$ and $0 < b < 1$ are parameters which determine the intensity and the type of competition among the larvae. An obvious question concerning this model is how the iteration behaves. Under which circumstances, i.e. for which values of the constants and the parameters, does the population develop in a certain way? Is there a cycle: big females → many eggs → poor conditions → few females → few eggs → good conditions → and so on? Does the iteration converge, and if so when? Or is there no pattern at all?

These questions will be studied in the following section.

3.3 Discussion of the fixed point iteration

To study the behaviour of the iteration we will first discuss the iteration function, f. Recall that

$$f(x) := bax\Big(1 - \frac{x}{v}\Big)^r \cdot h(x).$$

3.3.1 Properties of f

Note first that f has zeroes at $x = 0$ and $x = v$ and is positive in $(0, v)$. Furthermore, f is continuous on $[0, v]$ and differentiable on $[0, v)$. The derivative of f is given by

Lemma 3.3.1

$$f'(x) = \frac{f(x)}{x}\left(1 - r\frac{x}{v-x} + x\frac{h'(x)}{h(x)}\right), \tag{3.6}$$

for $x \in (0, v)$. We will also need f' at $x = 0$:

$$f'(0) = baM. \tag{3.7}$$

Proof:

$$\begin{aligned} f'(x) &= ba\left[\left(x \cdot r\left(1 - \frac{x}{v}\right)^{r-1} \cdot \left(-\frac{1}{v}\right) + \left(1 - \frac{x}{v}\right)^r\right) \cdot h(x)\right. \\ &\quad \left. + x\left(1 - \frac{x}{v}\right)^r h'(x)\right]. \end{aligned}$$

From this one obtains (3.7). If $x \in (0, v)$, f' simplifies to

$$\begin{aligned} f'(x) &= ba\left[\left(1 - \frac{x}{v}\right)^r \left(1 - \frac{xr}{v(1 - \frac{x}{v})}\right) \cdot h(x)\right. \\ &\quad \left. + x\left(1 - \frac{x}{v}\right)^r \cdot h'(x)\right] \\ &= \frac{f(x)}{x}\left(1 - r\frac{x}{v-x}\right) + f(x)\frac{h'(x)}{h(x)} \\ &= \frac{f(x)}{x}\left(1 - r\frac{x}{v-x} + x\frac{h'(x)}{h(x)}\right). \end{aligned}$$

□

3.3.2 Fixed points

In order to study whether the iteration converges or diverges one needs to know whether f has fixed points. Clearly, $x_0^* = 0$ is a fixed point of f. The following proposition states under which conditions there are others.

Proposition 3.3.1

- *If* $b \leq \frac{1}{M\alpha}$ *then* $f(x) < x$ *for all* $x \in (0, v]$. *Thus,* f *has precisely one fixed point* $x_0^* = 0$.
- *If* $b > \frac{1}{M\alpha}$ *then* f *has precisely two fixed points* $x_0^* = 0$ *and* $x^* \in (0, v)$.

Proof:
To prove the first part of the proposition suppose that $b \leq \frac{1}{M\alpha}$. Then for $0 < x \leq v$ one has $b\alpha(1 - \frac{x}{v})^r < b\alpha(1 - \frac{0}{v})^r = b\alpha \leq \frac{1}{M}$. Thus

$$\underbrace{b\alpha\left(1 - \frac{x}{v}\right)^r}_{\leq \frac{1}{M}} \underbrace{h(x)}_{<M} < 1$$

and hence $f(x) = b\alpha x(1 - \frac{x}{v})^r h(x) < x$.

To prove the second part of the proposition suppose that $b > \frac{1}{M\alpha}$. Then $f'(0) = b\alpha M > 1$. Using the marginal value theorem it follows that f has at least one fixed point in $(0, v)$: define y(x) := f(x)-x then because $f'(0) > 1$ there exists $\xi \in (0, v)$ for which $y(x) > 0$. Since $y(v) < 0$ and since y is continuous there must be an $x \in (\xi, v)$ for which $y(x) = 0$, i.e. for which $f(x) = x$.

It remains to show that there cannot be more than one fixed point. First define $g(x) := b\alpha(1 - \frac{x}{v})^r$. Then both, g and h, are positive and strictly monotonically decreasing functions on $(0, v)$ and a fixed point of f has to satisfy the equation

$$f(x) = x$$

or equivalently

$$g(x) \cdot h(x) = 1.$$

Now, suppose f has two fixed points, x_1^* and x_2^*, in $(0, v)$. Then one of the two will be smaller than the other. So suppose that $x_1^* < x_2^*$. Since g decreases strictly monotonically it follows that $g(x_2^*) < g(x_1^*)$. Since the product of g and h is constant this implies that $h(x_2^*) > h(x_1^*)$. This however, leads to a contradiction since h also decreases strictly monotonically, hence there can only be one fixed point in $(0, v)$.

□

In order to find an expression for the second fixed point, x^*, which lies in the interval $(0, v)$ one would have to solve the equation $f(x) = x$. Since we have kept h general we cannot give an explicit solution. It is however possible to describe qualitatively how variation of the parameters b and r influences the fixed point x^*. Note that low competition among larvae corresponds to large values of b and/or small values of r. Conversely, high competition among larvae corresponds to small values of b and/or large values of r.

Lemma 3.3.2
Suppose $b > \frac{1}{M\alpha}$, *i.e.* f *has a second fixed point* x^* *in* $(0, v)$.

(1) If one of the two parameters, r *and* b*, is kept fixed while the other one is varied such that the competition among larvae is decreased the fixed point* x^* *increases and vice versa.*

(2) If one of the parameters r *and* b *approaches infinity while the other one is kept fixed* x^* *approaches* 0.

Proof:
(1) :

In proposition (3.3.1) it has been shown that a fixed point of f has to satisfy the equation

$$g(x) \cdot h(x) = 1, \tag{3.8}$$

where $g(x) := b\alpha(1 - \frac{x}{v})^r$. Recall also that g and h are positive and strictly monotonically decreasing functions on $(0, v)$.

Now suppose that $x^* \in (0, v)$ is fixed point for some given values of b and r. Then increasing b (i.e. lowering the competition) and keeping r fixed will result in an increase of $g(x^*)$. $h(x^*)$ on the other hand does not change. For condition (3.8) to hold again the fixed point corresponding to the new values of b and r has to be larger than the previous one. The same is true if r is decreased (i.e. competition is decreased) and b is kept fixed. This proves part (1) of the lemma.

(2) :
Suppose b is fixed. If $r \to \infty$ then $f(x) = b\alpha x(1 - \frac{x}{v})^r(\frac{m-M}{v}x + M) \to 0$ $\forall x \in (0, v)$. Thus, $x^* \to 0$ since $x^* = f(x^*)$ and $x^* \in (0, v)$. Similarly, if r is fixed and $b \to \infty$ $f(x) \to 0$ and thus $x^* \to 0$.

□

3.3.3 Convergence/Divergence

A fixed point iteration converges locally towards a fixed point if the slope of the iteration function at this fixed point is greater than -1 and smaller than 1. It diverges if the slope at the fixed point is smaller than -1 or greater than 1.

Proposition 3.3.2
For $b \leq \frac{1}{M\alpha}$ the iteration converges for any starting point in $[0, v]$ towards $x_0^* = 0$.

Proof:
From equation (3.7) one obtains that $f'(0) < 1$ and hence the iteration converges locally towards $x_0^* = 0$.

Now, let $(x_g)_{g \geq 0}$ denote the iteration series. Then $(x_g)_{g \geq 0}$ decreases: from proposition 3.3.1 we know that $f(x) < x$ for all $x \in (0, v]$. Thus $x_{g+1} = f(x_g) < x_g$ for all $g \geq 0$ (if x_g was not already equal to 0).

Since $x_0^* = 0$ is the only fixed point of f and since f is positive this proves that the iteration converges towards $x_0^* = 0$ for any starting point in $[0, v]$.

□

In the case where f has two fixed points we find that

Proposition 3.3.3
If $b > \frac{1}{M\alpha}$ there is no convergence towards the fixed point $x_0^* = 0$ for any starting point in $(0, v)$.

Proof:
This follows directly from the fact that here $f'(0) = b\alpha M > 1$.

□

For the second fixed point the discussion is not so simple. If $x^* \in (0, v)$ is fixed point of f one obtains the slope of f at x^* by substituting $x^* = f(x^*)$ into equation (3.6):

$$f'(x^*) = 1 - r\frac{x^*}{v - x^*} + x^* \frac{h'(x^*)}{h(x^*)}. \tag{3.9}$$

Now, define $A := \frac{x^*}{v-x^*}$ and $B := x^* \frac{-h'(x^*)}{h(x^*)}$. Then this is equivalent to

$$f'(x^*) = 1 - rA - B. \tag{3.10}$$

A and B are both positive since $x^* \in (0, v)$ and since h is positive and h' negative. Thus, $f'(x^*) < 1$ for all possible values of r and b.

According to lemma 3.3.2 the fixed point x^* increases if one of the two parameters r and b is kept fixed and the other one is altered such that competition decreases. Note, that A increases if x^* increases. Whether B increases with x^* depends on h. If, for example, h is a line or if h'' exists and is negative B will increase with x^*. And in this case $f'(x^*)$ will decrease.

We will summarize the results obtained in this section in the following

Corollary 3.3.1
Consider the fixed point iteration derived in section 3.2.

- *For* $b \leq \frac{1}{M\alpha}$, $x^* = 0$ *is the only fixed point of the iteration function* f. *The iteration converges towards* $x_0^* = 0$ *for any starting point in* $[0, v]$.
- *If* $b > \frac{1}{M\alpha}$, f *has a second fixed point.*

  There is no convergence towards the fixed point $x_0^* = 0$ *for any starting point in* $(0, v)$.

  The second fixed point lies in $(0, v)$ *and decreases to* 0 *if one of the two parameters* r *and* b *is altered such that competition among larvae is increased while the other one is kept fixed.*

  The slope at the second fixed point is always smaller than 1. *If* h *is such that* $-x\frac{h'(x)}{h(x)}$ *is a monotonically increasing function the slope at the fixed point will increase if the fixed point decreases.*

  If one parameter is fixed and the other one varies this leaves us with three possibilities:

  (1) The iteration converges locally for all values of the varying parameter.

  (2) The iteration diverges for all values of the varying parameter.

  (3) The iteration converges locally for low competition and diverges for high competition.

3.3.4 Cycles

If an iteration diverges it does not mean that there is no cyclic pattern. In general iterations can produce extremely complex behaviours and there is no straight forward way of finding patterns like cycles. In this case, however, we are particularly interested in whether it can occur that every second generation lives in a high density environment while the generations in between live in a low density environment.

Consider the two series defined by

$$A_{g+1} := f(f(A_g)) \quad \text{and} \quad B_{g+1} := f(f(B_g)),$$

where $g = 0, 1, 2, \ldots$, $A_0 := N_0$ and $B_0 := f(N_0)$.

One can speak of a cycle if these two series converge and have different limits.

Note that the above equations again define fixed point iterations this time with iteration function $F(x) := (f \circ f)(x)$ for $x \in [0, v]$. So for a cycle to exist F would have to have at least two fixed points. Note that zero is a fixed point of F. We are, however, not interested in this fixed point because in reality it would correspond to the death of the whole population and so certainly not to a cycle. We will therefore study F on the interval $(0, v)$.

From the properties of f we can obtain some information about F. Denote the maximum value attained by f by f_{max}.

- Zeroes:

  If $f_{max} < v$, the zeroes of F are the same as the zeroes of f. This is because $F(x) = 0$ is equivalent to $f(f(x)) = 0$ which is only satisfied if $x = 0$ or $x = v$.

  If $f_{max} \geq v$ then $F(x) = 0$ for $x = 0$ and $x = v$ and additionally for all $x \in (0, v)$ with $f(x) \geq f_{max}$.

- Fixed points:

  F has at least one fixed point. It coincides with the fixed point of f since $F(x^*) = f(f(x^*)) = f(x^*) = x^*$. Unless F has other fixed points in $(0, x^*)$ cycles cannot occur.

  Note also that $F'(x) = f'(f(x))f'(x)$ and hence $F'(x^*) = (f'(x^*))^2$.

3.4 Simulation

We will now plot f and F for varying parameters. The identity line will be displayed in the plots to indicate where the fixed points lie.

This section will be divided into three parts. The first part will discuss f and deduce from it information about stable and unstable population equilibria predicted by the model. The second part will discuss F and the occurrence of cycles of the form described in the previous section. Finally, in part three we will run the iteration for 30 generations and plot the number of surviving females and the clutch sizes they produce.

In all cases α, the parameter that determines the proportion of the surviving adults that are females, will be set to $\frac{1}{2}$.

- **density-dependent fecundity:**

  For h we choose the line

  $$h(x) := \frac{M-m}{v}x + m.$$

  This means that for low densities of larvae in the environment the clutch size produced by the resulting adult females is large but does not exceed M. For high larval density the resulting clutch size is small but never drops below m.

  Note also that $-x\frac{h'(x)}{h(x)}$ is an increasing function of x. This condition was needed in Corollary (3.3.1).

  The following plots were generated for $M = 100$, $m = 50$ and $v = 55000$.

- **Competition:**

  Given N_g, the number of eggs in the environment in generation g, the number females that survive to adulthood is given by

  $$F_g(N_g) := \alpha \cdot bN_g \left(1 - \frac{N_g}{v}\right)^r.$$

  We will choose the integral over this curve as a measure for the level of larval competition to which the curve corresponds. Neglecting the constant factor α, gives

  $$C(b,r) := \int_0^v bx(1-\frac{x}{v})^r dx = \frac{bv^2}{(r+1)(r+2)}. \tag{3.11}$$

  The diagrams shown in part one and two of this section correspond to $b_1 = 0.015, b_2 = \frac{1}{M\alpha} = 0.02, b_3 = 0.042, b_4 = 0.06, b_5 = 0.1$ and $b_6 = 0.2$.

  The curves shown in each of the diagrams correspond to the competition levels

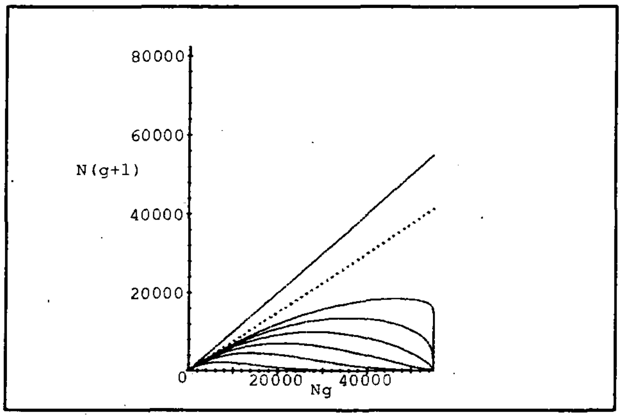
  $C_i := C(0.042, \frac{i}{12-i})$ for $i = 1, 2, ..., 11$.

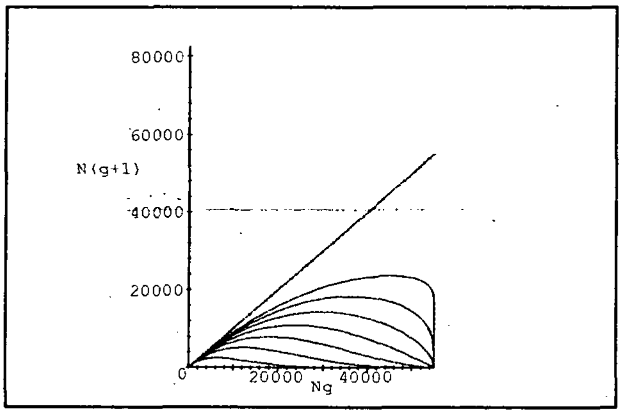
  For each curve the parameter r_i is obtained by solving $C(b_i, r_i) = C_i$, which leads to

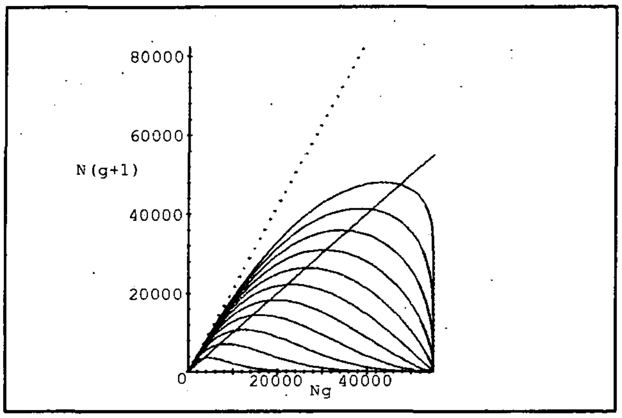
  $r_i := -\frac{3}{2} + \sqrt{\frac{1}{4} + \frac{bv^2}{C_i}}$.

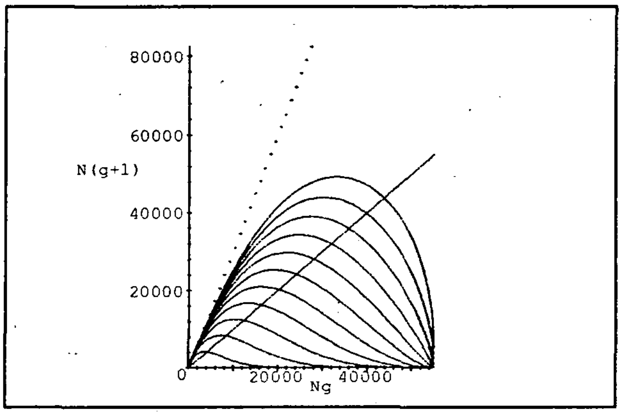
3.4.1 Population equilibria

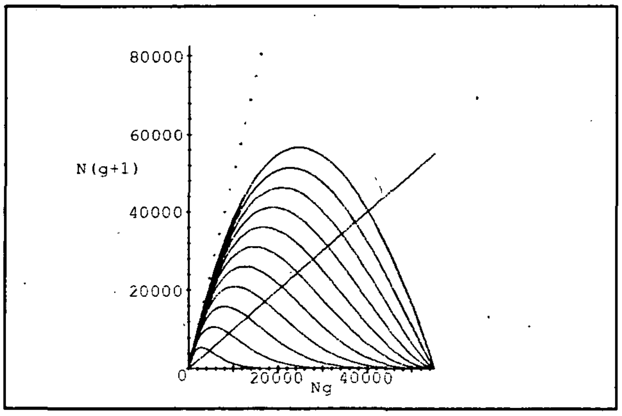
In the following plots the solid line is the identity line and the dotted line is the tangent to f at $x = 0$.

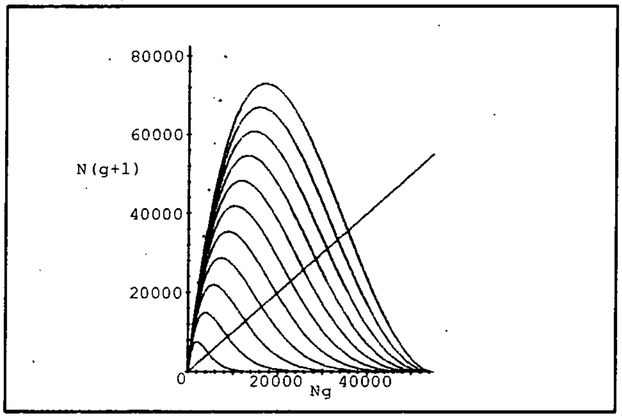





80000
60000
40000
20000
N(g+1)
0
20000
40000
Ng


80000
60000
40000
20000
N(g+1)
0
20000
40000
Ng


80000
60000
40000
20000
N(g+1)
0
20000
40000
Ng



Discussion:

- $b \leq \frac{1}{M\alpha}$

  The first two plots are examples of the case where the model predicts the death of the whole population. There is no fixed point other than $x^* = 0$.

- $b > \frac{1}{M\alpha}$

  The other four plots are examples of the case where the model predicts the existence of a non-zero population equilibrium.

  Here, the slope of f at $x = 0$ is larger than 1 and hence the population will not die out unless N_g moves into an interval inside $(0, v)$ on which $f(N_g) = 0$. If that is the case the iteration will jump to $x = 0$ in the next step. The model then predicts the death of the population since $x^* = 0$ is an equilibrium point. But this equilibrium is unstable and so in reality one would expect a scenario where the population size drops to almost zero and then recovers again.

  The plots show that if b is fixed and the level of competition is decreased the population equilibrium will increase and in general is expected to switch from a stable equilibrium into an unstable one. If b is very large the number of offspring is optimal for relatively low densities of eggs. The plots predict for this case that the equilibrium is generally unstable. Conversely, if b small the number of offspring is optimal for relatively high densities of eggs and the plots predict that the equilibrium is generally stable.

  If b is increased and r is adjusted such that C, the integral over f, stays fixed the plots predict that for all C the population equilibrium will switch from a stable equilibrium into an unstable one.

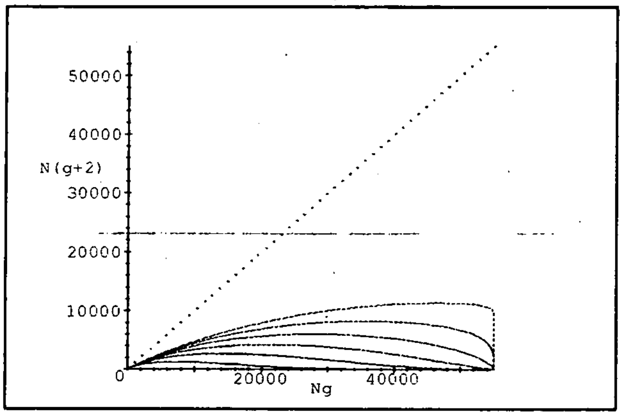
The following summarizes these results:

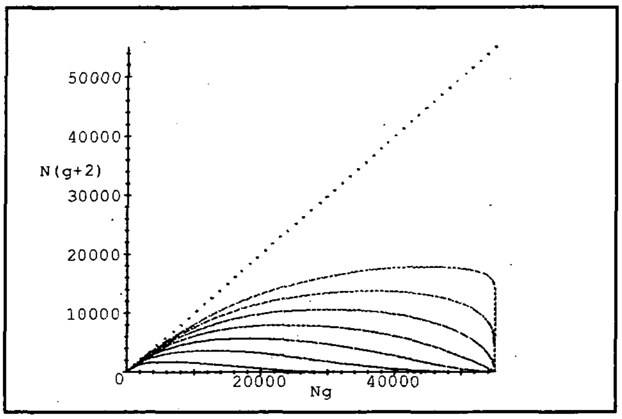
Prediction 3.1
Competition among larvae has a stabilizing effect on population size.

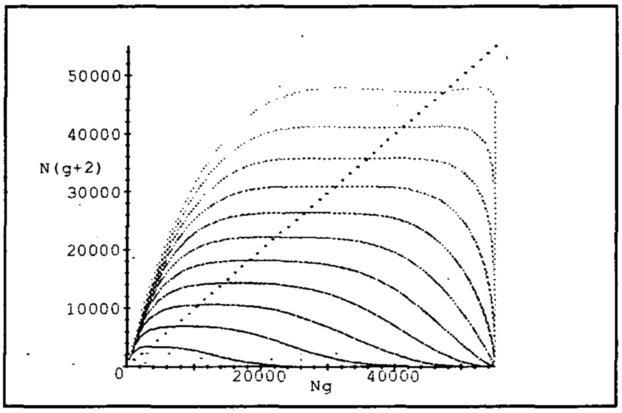
Prediction 3.2
Let N^* denote the density of eggs that results in the maximum number of offspring. Then higher N^* leads to larger and to more stable population equilibria. Conversely, lower N^* implies a higher likelihood of chaotic behaviour and smaller equilibria.

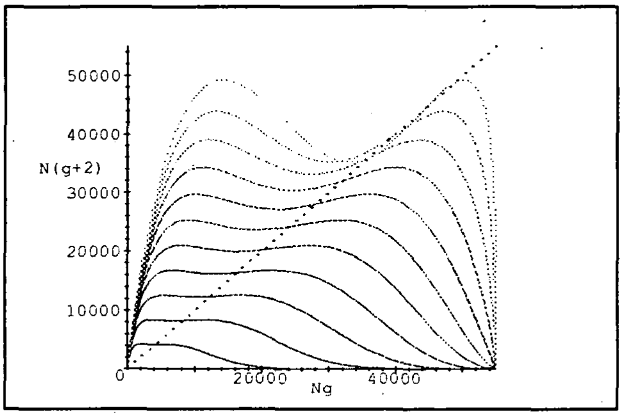
3.4.2 Cycles

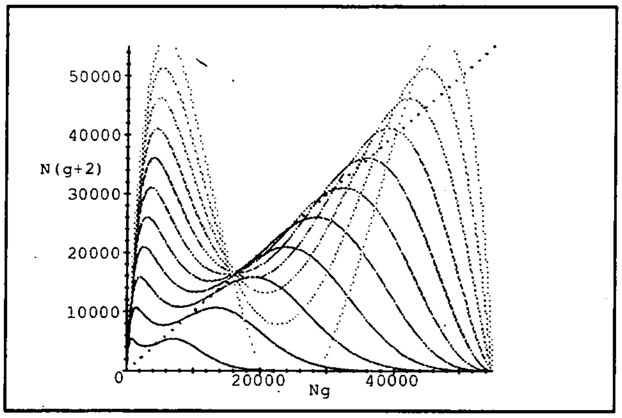
The following six diagrams show F for the same values of b and r that were chosen for the plots of f above. As before the identity line is also included in the diagrams to indicate where the fixed points lie.

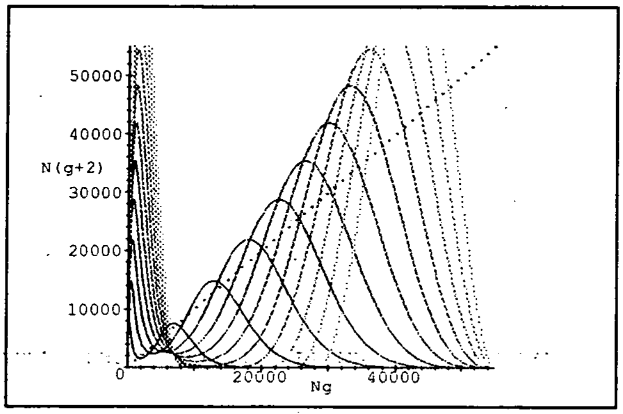



50000
40000
N(g+2)
30000
20000
10000
0
20000
Ng
40000


50000
40000
N(g+2)
30000
20000
10000
0
20000
Ng
40000


50000
40000
N(g+2)
30000
20000
10000
0
20000
Ng
40000





Discussion:

- $b \leq \frac{1}{Ma}$

  It was shown in the proof of proposition (3.3.2) that in this case the series $(N_g)_{g \geq 0}$ decreases to zero for any starting point in $(0, v)$. The first two plots confirm this: F has no fixed points other than $x = 0$ and hence there are no cycles.

- $b > \frac{1}{Ma}$

  The other four plots indicate that F can have two additional fixed points which lie on either side of x^* which is fixed point of both, f and F. The two additional fixed points exist only when x^* corresponds to an unstable population equilibrium. This means that cycles only occur when the iteration given by f diverges. Furthermore it implies that if the iteration given by f converges locally towards x^* it in fact converges for any starting point in $(0, v)$.

As parameters vary, cyclic behaviour occurs when the iteration given by f switches from convergence to divergence, i.e. when $|f'(x^*)|$ is only slightly larger than 1. In this case according to the plots the two series (A_g) and (B_g) converge. The two points between which the population will alternate are the two fixed points of F which lie on either side of x^*. This is because $F'(x^*) = (f'(x^*))^2$ and hence $F'(x^*) > 1$ if $f'(x^*) > 1$.

Prediction 3.3
As parameters vary alternating population sizes occur when convergence switches into divergence or more precisely when the absolute value of the slope of f at x^* is just above 1.

3.4.3 Simulation of 30 subsequent generations

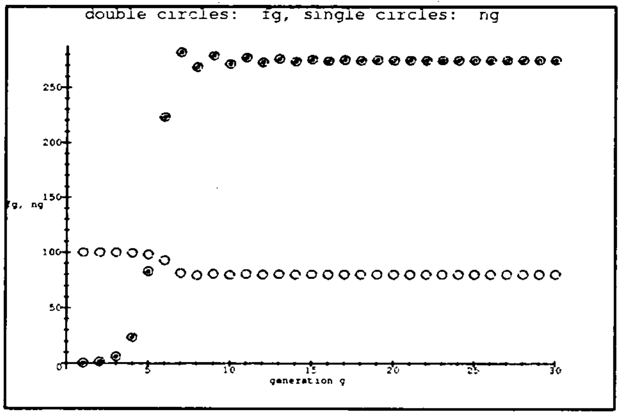
We will now run the iteration for 30 generations and plot the number of surviving females and the clutch sizes that these females produce.

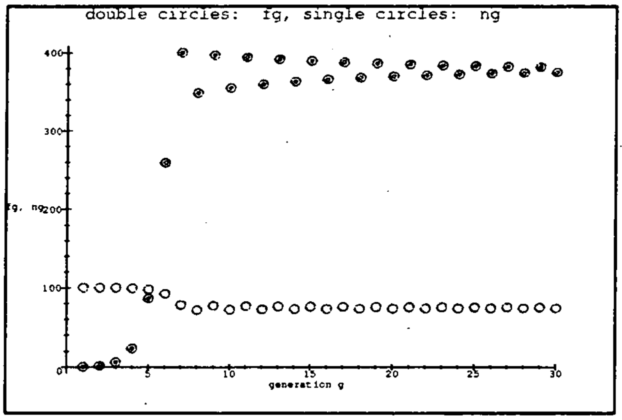
The following plots were generated for

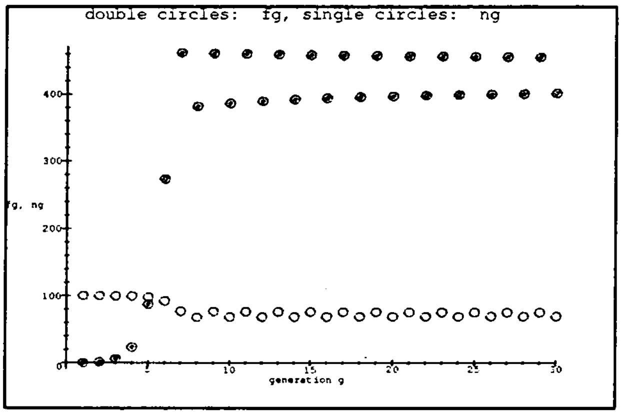
$b = 1/(M * \alpha) + 3/50 * (1 - 1/(M * \alpha))$

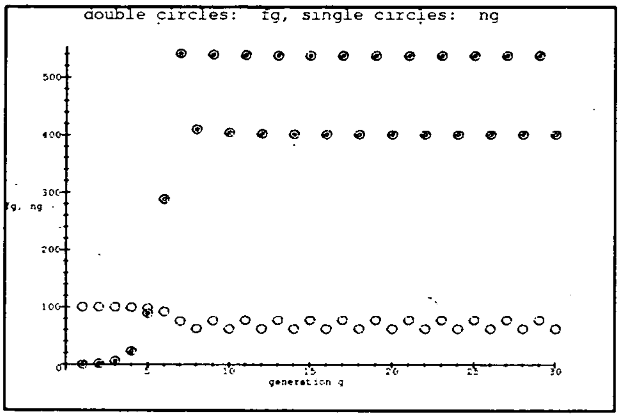
and

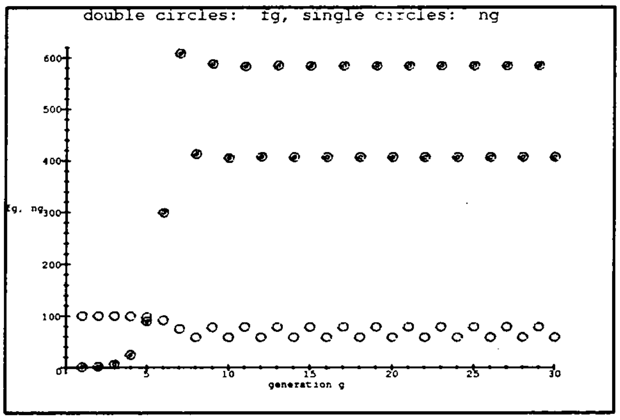
r = 3, 2.25, 1.5, 1.25, 1, 0.83333, 0.66666, 0.5, 0,33333

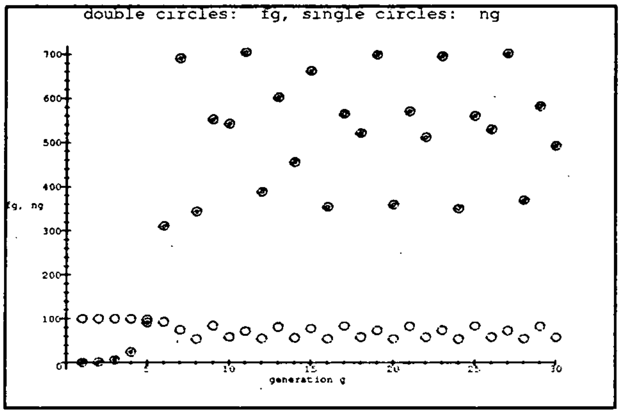


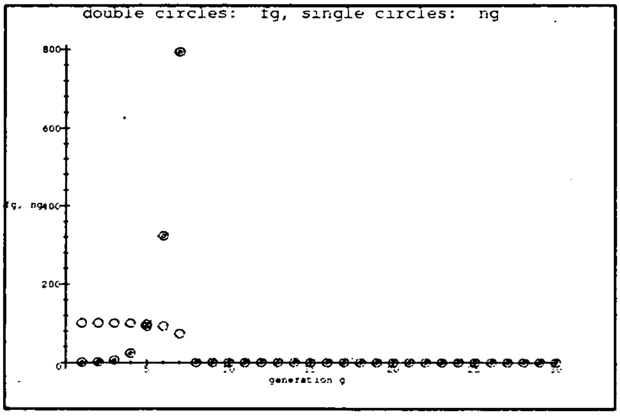

double circles: fg, single circles: ng
400
300
fg, ng
200
100
0
5
10
15
20
25
30
generation g

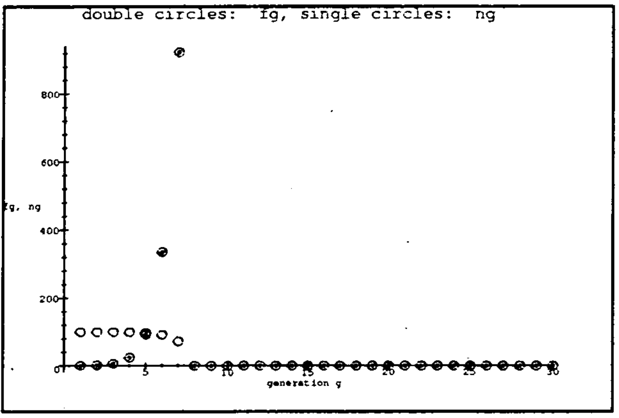

double circles: fg, single circles: ng
400
300
fg, ng
200
100
0
5
10
15
20
25
30
generation g


double circles: fg, single circles: ng
500
400
300
fg, ng
200
100
0
5
10
15
20
25
30
generation g


double circles: fg, single circles: ng
fg, ng
600
500
400
300
200
100
0
5
10
15
20
25
30
generation g


double circles: fg, single circles: ng
fg, ng
700
600
500
400
300
200
100
0
5
10
15
20
25
30
generation g


double circles: fg, single circles: ng
fg, ng
800
600
400
200
0
5
10
15
20
25
30
generation g



Discussion:

- Plots one to three show convergence.
- Plots four to six show cycles.
- The last three plots show what can happen when the two series (A_g) and (B_g) do not converge. The population size varies unpredictably or increases beyond the maximum size sustainable by the environment and then collapses into 0. As already mentioned above $x = 0$ is an unstable equilibrium and so rather than dying out one would expect the population to drop to almost zero size and then to recover again.

3.5 Discussion

The model derived in this chapter was designed to study the effects of density-dependent fecundity on population fluctuations.

It was originally motivated by experiments conducted on *Trichoprosopon digitatum* (Diptera: culicidae) by Dr. Tom Sherratt. The collected data (unpublished) shows that high larval density or low food level result in longer development times and smaller adults. Green (1989) found that larger females are more successful in surviving to reproduce and produce larger first clutches compared to smaller females. This could result in density-dependent feedback.

The assumptions that were made to keep the model mathematically simple are, however, not suitable for *Trichoprosopon digitatum*. The assumptions were as follows

(a) the carrying capacity of the larval habitat is constant

(b) females lay one clutch only

(c) females do not retain eggs

(d) adults do not affect the living conditions of the larvae

(e) adults of different generations oviposit in discrete time intervals

(f) clutch size is constant within generations

(g) larval survival chances and adult fecundity decrease with increasing larval density

(e) and (f) are not satisfied and (a) seems to be an invalid assumption because breeding conditions depend largely on rainfall (see Green (1989)). In fact Barr *et al.* (1986) found a seasonal difference in the number of eggs in *Culiseta incidens* rafts in California which they attributed to variation in female size with season: abundant rainfall in spring results in a low density environment for the larvae and consequently shorter development times and larger adults, whereas during the dry summer living conditions among larvae are poor and result in smaller adults. This means that effects that might be caused by density-dependent feedback could also be caused by seasonal changes in the environment.

The model developed and analyzed in this chapter predicts a general trend for species for which the assumptions are reasonable: it predicts a transition from stable to cyclic to chaotic population fluctuations as competition levels among larvae decrease. The above assumptions are relatively limiting. However, if the assumptions are partially satisfied the predicted behaviour could also occur in a more complex system as one of many factors that determine population size variation.

Chapter 4

Variation in egg size and dependency of survival rate on clutch size in *Pieris brassicae* (Lepidoptera: Pieridae)

4.1 Introduction

The first of the two experiments described in this chapter was conducted to investigate possible variation of egg size in *Pieris brassicae* and to estimate larval growth rate. A second experiment studies the relationship between larval density and survival rates. Hubbard (1972,1977) found that the main causes of larval mortality in *P. brassicae* were rainfall in the first three instars and bird predation and parasitism by *Cotesia glomerata* (L.) in the fourth and fifth instar. Kristensen (1994) found that 96.9% of the total mortality during the larval stages of *P. brassicae* was due to predation and that if predation was restricted to arthropods, survival rates were increased eleven times. Le Masurier (1994) studied density-dependent mortality in *P. brassicae*. He found no density-dependent mortality during the first three instars. For the fourth and fifth instar he showed that parasitism increased with larval density but he found no evidence of a percentage increase of parasitism with density. In his study bird predation was excluded. In this chapter survival rate will be studied in a laboratory experiment where predation , parasitism and rainfall are excluded. The experiment will study the effect of larval density on competition for food and resulting survival rate.

4.2 Methods

Two separate laboratory experiments were conducted in summer 1996. For both *P. brassicae* eggs ordered from a butterfly farm were used. The eggs arrived in clutches, three of which were used in experiment 1, the remaining ten were used in experiment 2. Eggs and caterpillars were kept in a laboratory room under room temperature without direct sunlight. Temperatures were noted every day, unfortunately at irregular times. A third experiment, intended to study the oviposition behaviour of *P. brassicae* females, failed because too few adults were obtained from experiments 1 and 2.

Experiment 1

The eggs from three clutches were carefully separated using a sharp scalpel. All damages to the eggs, already existent or due to separating them, were noted. The width and height of each egg were measured under a microscope. The measured egg was then put into a labeled petri dish. The hatching of the eggs had to be monitored under the microscope. Since there were altogether 91 eggs it was not possible to note exact hatching times. All eggs hatched on one day and it was noted whether they hatched in the morning, the early afternoon or the evening. As shortly after hatching as possible the length and width of the caterpillars were measured under the microscope.

The caterpillars were fed on white cabbage. It was ensured that they were provided with plenty of food and that the quality of the cabbage was roughly the same for each caterpillar. The cabbage was replaced every three to four days and at the same time the petri dishes were cleaned. Every four days the caterpillars were weighed. Three measurements were taken for each caterpillar accurate to five decimal places.

The days on which the caterpillars pupated were noted and pupae were weighed again taking three measurements accurate to five decimal places for each pupa. The sizes of the pupae were measured in the following way: the length was measured omitting the last four segments of the abdomen since the pupae tended to move that part of their body. The width was measured where the pupa was widest.

Finally, the day of emergence was noted for each pupa. Adults were not measured since there were too few of them to give data of any statistical significance.

Experiment 2

After hatching, the caterpillars of ten clutches were divided into groups of 5, 10, 20 and 40 caterpillars and put into plastic tubs. In this fashion five tubs for each density were set up. All caterpillars of a tub were from the same original clutch. One additional tub was set up, containing the 39 caterpillars that were left over.

The caterpillars were fed on white cabbage. The leaves were weighed and each tub was provided with the same amount of cabbage. Every three to four days the tubs were cleaned out and provided with fresh cabbage. The amount of cabbage replaced in this way was kept constant throughout the experiment (leaves were weighed, the weight of the leaves for each tub was between $11.45g$ and $11.54g$).

The caterpillars were counted every three days for the first ten days, then every day. They were weighed but unfortunately not often enough to provide sufficiently many sample points for a meaningful statistical analysis.

As in experiment 1 the pupae were weighed and measured and pupation days and emergence days were noted.

During experiment 2 it was observed that in the case of density 5 the larvae never ate all the food they were given. They did not appear to experience any competition for food. In the case of the three higher densities, however, competition was observed and occurred irregularly whenever the larvae had finished their food.

4.3 Results

4.3.1 Variation in egg size

To analyze variation in egg size the approximate volume of the eggs was calculated from the widths and heights measured in experiment 1 using the formula for the volume of an ellipsoid

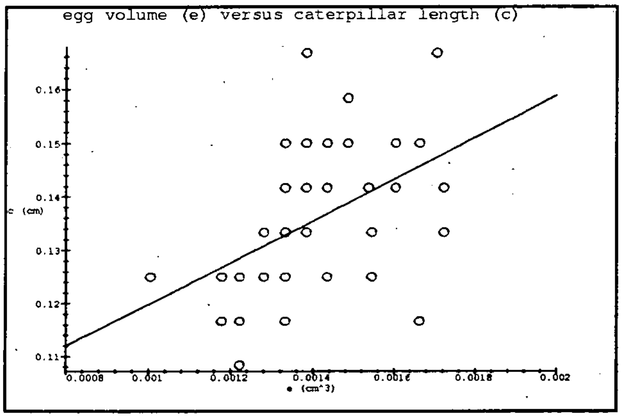
$$V(egg) \approx \frac{4}{3}\pi a^2 b,$$

where a is half of the width and b half of the height of the egg.

To determine whether there was significant variation in egg sizes the standard deviation of the egg volumes was calculated. This was then multiplied by 100 and divide by the mean of the egg volumes to transform it into percentages of the mean. The result is

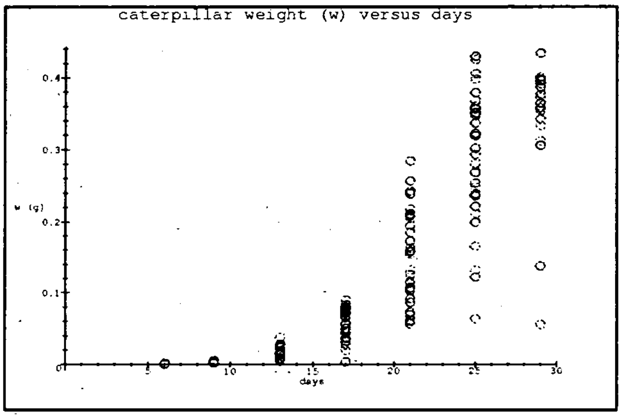
$$\begin{pmatrix}\text{standard devia-}\\ \text{tion of egg vol-}\\ \text{umes}\end{pmatrix} \cdot \frac{100}{\text{mean of egg volumes}} = 12.14474838\%.$$

To find out whether egg size and caterpillar size were related, caterpillar length was plotted against egg volume (no variation in caterpillar width was observed) and a line was fitted to the data using the method of least squares. The correlation coefficient shows that there is a highly significant positive correlation between caterpillar size and egg size.



The regression line is given by y = 39.03420342x+.0807990133. The observed correlation coefficient is given by $r_o = .4615610148$ ($p = .0035355332$, $N = 38$, where $p = P_N(|r| \geq |r_o||H_0)$, H_0: the two variables are uncorrelated).

4.3.2 Growth rate



The above diagram shows a scatter plot of the weights of the caterpillars versus days, where triple measurements have been converted into their means.

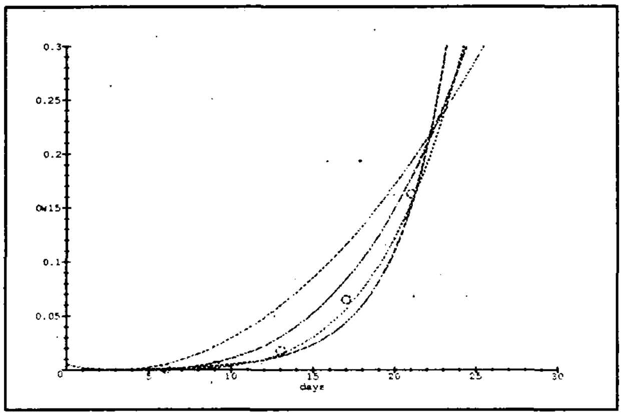
To determine a measure for the growth rate of each of the caterpillars in experiment 1, the shifted monomials $y = a(x-3)^n$, where $n = 2, 3$ or 4, were fitted to the data. The reason for shifting them by 3 is that all caterpillars hatched on day 3 of the experiment. As a fourth type of curve, exponentials $y = ae^{bx}$ were fitted to the data points.

In each of the above cases the residual sum of squares (RSS) and the standard deviation were calculated. Thus, for each case two lists were obtained, one containing the RSSes and the other containing the standard deviations of all caterpillars. To compare the fit of the curves the mean and the variance of these lists were computed, giving for each case a measure for the mean deviation of the data from the fitted curves and a measure for how much the deviation varied between caterpillars as well as a measure for the mean error spread and a measure for how much the error spread varied between caterpillars. The following table shows the resulting values.

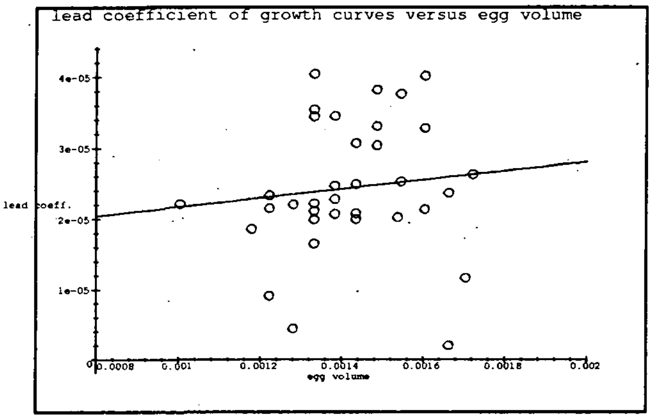
Assessment of fit				
	RSSes		std dev	
	mean	variance	mean	variance
parabola	.74931618e-2	.20039635e-4	.28985390e-1	.11293023e-3
cubic	.26618375e-2	.11118980e-4	.16586731e-1	.93887417e-4
quartic	.52956279e-2	.55782098e-4	.21403088e-1	.22890037e-3
exponential	.91163609e-1	.70369928e-2	.99622488e-1	.26519781e-2

The values show that the cubic monomials fit the data best and that the worst fit is achieved by exponentials even though in the latter case two parameters were optimized instead of one.

The following diagram shows quadratic, cubic, quartic (dashed lines) and exponential (solid line) regression curves for one particular caterpillar.

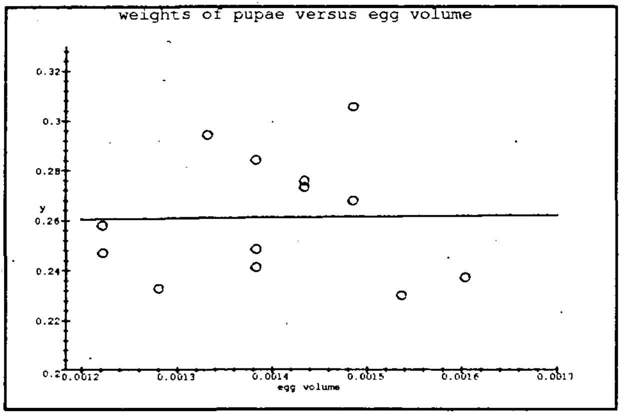


The following is a scatter plot of the lead coefficients of the cubic regression curves versus egg volume. It shows no significant correlation between egg volume and growth rate.



The regression line is given by y := .0063705220x+.0000153323. The observed correlation coefficient is given by $r_o = .1112724585$ ($p = .5245351809$, $N = 35$, where $p = P_N(|r| \geq |r_o||H_0)$, H_0: the two variables are uncorrelated).

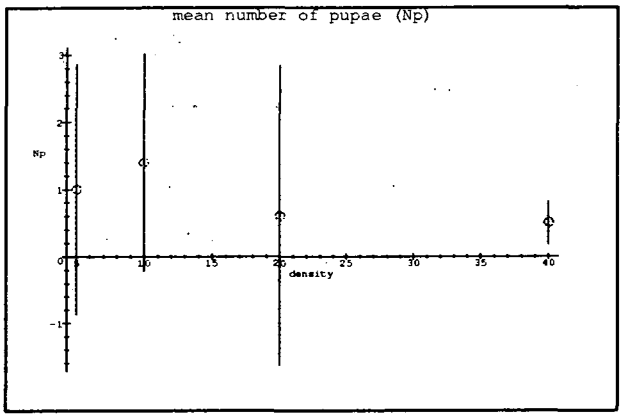
Nor is there a significant correlation between egg volume and pupal weights:



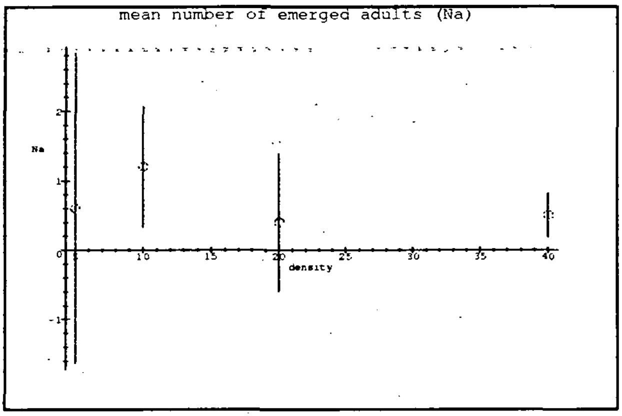
The regression line is given by y := 3.191397909x+.2567111859. The observed correlation coefficient is given by $r_o = .0151807288$ ($p = .9607425536$, $N = 13$, where $p = P_N(|r| \geq |r_o||H_0)$, H_0: the two variables are uncorrelated).

4.3.3 Survival rate

The following plots show (1) the number of pupae versus density (averaged over all treatments of that density) including 95% confidence intervals, (2) the number of emerged adults versus density (averaged over all treatments of that density) including 95% confidence intervals. Listed underneath each plot are the results of a one-way analysis of variance.



ANOVA: d1 = 3, d2 = 17, F = .7607083034, p = .4684909801.

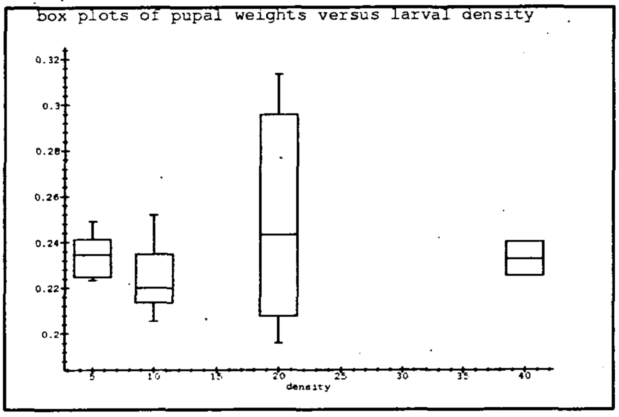


ANOVA: d1 = 3, d2 = 17, F = .7581254724, p = .4671068349.

In both cases the p-value is much greater than 0.05 and hence the null hypothesis that there is no systematic difference among treatments cannot be rejected. Thus, the means calculated for different densities are not significantly different.

The most likely explanation for this result is that too few larvae pupated successfully and even fewer emerged as adults. Out of the 375 larvae that were used in experiment 2 only 18 pupated and 14 emerged as adults. This provided not sufficient data to estimate any functional relationship between larval density and survivorship. The same is true for the weights of the pupae.

The following diagram shows box plots of the weights of the pupae versus density. Listed underneath the plot are the results of a one-way analysis of variance. Because of the lack of sufficient sample points pupae of different replicates of one density have not been distinguished.



ANOVA: d1 = 3, d2 = 13, F = .65645836744, p = .40685061452.

4.4 Discussion

The data collected in experiment 1 shows that variation in egg size is significant. It is, however, difficult to estimate to what an extent measurement errors, for example due to dents in the eggs, affect the results. The table in section 4.3.2 shows that larval growth is approximated very well by cubic monomials. Plots of the growth curves and the data confirm that cubic monomials achieve an extremely good fit. The growth rate, measured by the lead coefficient of the cubic regression curve, is not related to the egg volume. This suggests that the ability to accumulate resources through feeding is not significantly influenced by the resources provided in the egg. This raises the question of whether variation of egg size in *P. brassicae* might be due to a physiological inability to control the precise size of an egg. If so could the observed variation be merely a random effect rather than a behaviour that is selected for because of associated benefits?

Experiment 2 has to be interpreted with caution: due to being fed every three to four days rather than every day, the larvae reared in the densities 10, 20 and 40 experienced irregular starvation periods. It was observed that larvae in a tub had eaten all the food one or two days after they had been fed and hence starved for one or two days until new food was provided. Even though starvation can occur in *P. brassicae* when larvae migrate after depletion of the food plant, in the experiment starvation was caused artificially and, in the case of the two highest densities, occurred too early to reflect any realistic effect. Due to the high mortality of the larvae the data obtained from experiment 2 was not sufficient to allow any conclusions about a functional relationship between larval mortality and rearing density. To estimate a functional relationship the experimental design would have to be improved by increasing the number of replications and by ensuring a more constant food supply. The results found by Kristensen (1994) and Le Masurier (1994) (see the beginning of this chapter) suggest at best a weak influence of larval density on survivorship. Unfortunately, the author was not aware of these findings at the time of the experiment. To estimate a functional relationship between larval mortality and rearing density a species should be chosen for which a correlation between density and survivorship has been shown.

Chapter 5

Conclusion

In the second chapter of this thesis I have studied the approach to insect oviposition behaviour via mathematical optimization models, focusing on the fitness measure that is used in these models. In a third chapter I have studied an iterative model that investigates the effect of density dependent fecundity on population fluctuations.

In chapter 2 a further discussion of the model given by equation (2.1) showed that, assuming an increasing mortality risk, a female should invest decreasing amounts of reproductive resources with increasing age. This holds in the case where the female can vary both, the sizes of the clutches she lays and the sizes of her eggs, as well as in the case where she can vary only one of the two variables.

In the second part of chapter 2 a different measure for a female's fitness was discussed, the main idea being that it should be optimal for a female to maximize the number of her offspring but at the same time to ensure that her offspring will be as successful as possible. It was suggested that the success of the offspring should be expressed in terms of the amount of energetic resources that they can acquire as larvae since this would: • reduce the recursive definition of fitness to a non-recursive measure for fitness, • eliminate the conflict occurring between parent and offspring when the number of grandchildren is optimized, • allow to include realized fecundity rather than the fecundity measured in laboratory experiments, • be independent of the offspring's oviposition behaviour and • allow for the larvaes' sex to be included.

To build a new model based on these ideas further work would be needed to determine precisely what benefits are associated with larval energetic resource gain. The functional relationship between larval resource gain and realized fecundity would be of particular interest. More work on realized fecundity rather than fecundity measured in terms of the number of offspring produced by a female under laboratory conditions would also be needed.

In Chapter 3 a mathematical model was developed to study the effect of density-dependent adult fecundity on population fluctuations. Several assumptions were made to keep the model mathematically simple enough to allow some analytic results: (a) the carrying capacity of the larval habitat is constant, (b) females lay one clutch only, (c) females do not retain eggs, (d) adults do not affect the living conditions of the larvae, (e) adults of different generations oviposit in discrete time intervals, (f) clutch size is constant within generations and (g) larval survival chances and adult fecundity decrease with increasing larval density.

The model predicts a general trend for species for which the assumptions are reasonable: it predicts a transition from stable to cyclic to chaotic population fluctuations as competition levels among larvae decrease. This would essentially mean that larval competition has a stabilizing effect on population fluctuations. The above assumptions are relatively limiting. However, if the assumptions are partially satisfied, the predicted trend could also occur in a more complex system as one of many factors that determine population size variation.

Acknowledgements

I would like to thank my supervisor Dr. Tom Sherratt for always being very enthusiastic about any idea I presented to him and for supervising me remotely. I would also like to thank the College of St. Hild and St. Bede and in particular Dr. Armitage for supporting me.

I would like to thank the free software community, in particular the Linux community, for making software that is efficient, reliable and a pleasure to use.

Above all I thank my husband Sengan for all the encouragement he gave me and for greeting many of my theories with profound scepticism.

Appendix A

A C++ - program for chapter 2

```
//Offspring optimization:
//   Fctclass_PFdd pf = &Fctclass::clutchoffspring; (?)
//   pstagereturn = &Fctclass::stageoffspring;

//Resource/Grandchildren optimization:
//   Fctclass_PFdd pf = &Fctclass::clutchgain;
//    pstagereturn = &Fctclass::stagereturn;

//*******************************************************
//*******************************************************

#include <iostream.h>
#include <stdio.h>
#include <ctype.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

//*******************************************************
//CLASSES and TEMPLATES

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
class Stagetable {
  int k;
```

```
public:
  int* State;
  int* Decision;
  double* Cumreturn;
// constructor:
  Stagetable(int userk);
// destructor:
  ~Stagetable(void) { delete[] State;
                      delete[] Decision;
                      delete[] Cumreturn; }
};

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
class Strategy {
  int k;
public:
  int* Decisions;
  double* Stagereturns;
//constructor:
  Strategy(int userk);
//destructor:
  ~Strategy(void) { delete[] Decisions;
                    delete[] Stagereturns; }

};

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
class Fctclass {
  int N;
  int k;
  double* p;                 // survival probabilities
public:
  double* parameters;        // for clutchreturn fct
//constructor:
  Fctclass(double* programinput);
//destructor:
  ~Fctclass(void) { delete[] p;
                    delete[] parameters; }
  void probfct1(double pee);
  void probfct2(void);
  double clutchoffspring(double x);
  double clutchgain(double decision);
  double resourcetransf(double decision);
  double T1(double x);
  double T2(double x);
  double T3(double x);
```

```
  double T4(double x);
  double T5(double x);
  double T6(double x);
  double T7(double x);
  double stageoffspring(int stage, int decision);
  double stagereturn(int stage, int decision);
  int resultingstate(int state, int decision);
  int getindex(int lambda);




};
//*********************************************************
//TYPEDEFS

typedef void (Fctclass::*Fctclass_PFvv)(void);
typedef int (Fctclass::*Fctclass_PFid)(double);
typedef double (Fctclass::*Fctclass_PFdd)(double);
typedef double (Fctclass::*Fctclass_PFdii)(int, int);
typedef int (Fctclass::*Fctclass_PFiii)(int, int);
typedef int (Fctclass::*Fctclass_PFii)(int);

//*********************************************************



//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
class Toolclass {
  int k;
  int Lack;
  Fctclass* pFcts;
  Stagetable* Stagetables;   /* to be modified by
"laststage" and by
"findoptdecision" */
  int optdecision;           // to be initialised by
  double optcumreturn;       //        constructor

  Fctclass_PFdd pclutchoffspring;
  Fctclass_PFdd presourcetransf;
  Fctclass_PFdii pstageoffspring;
  Fctclass_PFdii pstagereturn;
  Fctclass_PFiii presultingstate;
  Fctclass_PFii pgetindex;
public:
//constructor:
```

```
  Toolclass(int userk, int Lack_from_main,
    Fctclass& Fcts_from_main);
//destructor:
  ~Toolclass(void) { delete[] Stagetables; }
//accessing functions:
  void laststage(int state);
  void findoptdecision(int stage, int state);
  void printStagetable(int which);
  void optimise(void);
  void makeStrategy(Strategy* pOptstrategy);
  void Toolclass::printStrategy(Strategy* pOptstrategy);
  void printinfo(Strategy* pOptstrategy);
};




//**********************************************************
//FUNCTION PROTOTYPES

//----------------------------------------------------------
// functions that handle user input:
int strisnumber(char* s);
void inputhandler_Nkptd(int* userinput_Nkptd,
                             int argc, char** argv);
double readfloat(void);

//----------------------------------------------------------
//non-member-functions for dynamic programming process:
int Lackfct(Fctclass& Fcts, Fctclass_PFdd pf, double v);

//----------------------------------------------------------
// function for general use:
int min(int a, int b);
int closestint(double x);


//**********************************************************
//**********************************************************
int main(int argc, char* argv[]) {
  int i,j;

  // inputhandler:
  int userinput_Nkptd[5];
  inputhandler_Nkptd(userinput_Nkptd, argc, argv);
```

```
// defines array which contains p - values
double pees[3];
pees[0] = 0.99; pees[1] = 0.93; pees[2] = 0.5;
// defines array which contains sigma - values
double sigmas[5];
sigmas[0] = 3.0; sigmas[1] = 1.5; sigmas[2] = 1.0;
sigmas[3] = 0.66666; sigmas[4] = 0.33333;

//.....................................................
// executes optimisation for all p's and all sigma's

for(i=0; i<=2; ++i) {
  printf("\\hline\n");
  printf("%g ", pees[i]);
  for(j=4; j>=0; --j) {
    printf("&%g ", sigmas[j]);


    //. . . . . . . . . . . . . . . . . . . . . . . . .
    // creates array to replace userinput
    double programinput[5];
    programinput[0] = double(userinput_Nkptd[0]);  //N
    programinput[1] = double(userinput_Nkptd[1]);  //k
    programinput[2] = pees[i];                      //pee
    programinput[3] = sigmas[j];                    //sigma
    programinput[4] = 0.0;                          //no data to be read in


    // create set of fcts to be used in optimisation
    // process
    Fctclass Fcts(programinput);

    // calculate Lack-clutch-size
    double v = Fcts.parameters[1];
    Fctclass_PFdd pf = &Fctclass::clutchgain;
    int Lack = Lackfct(Fcts, pf, v);
    printf("&%d ", Lack);

    // build strategy object
    Strategy Optstrategy(userinput_Nkptd[1]);
    Strategy* pOptstrategy=&Optstrategy;

    // create optimiser, run optimisation and
    // make strategy
    Toolclass Optimiser( userinput_Nkptd[1], Lack, Fcts);
```

```
//change this in inputhandler!
      Optimiser.optimise();
      Optimiser.makeStrategy(pOptstrategy);
      //Optimiser.printStrategy(pOptstrategy);
      Optimiser.printinfo(pOptstrategy);
      //. . . . . . . . . . . . . . . . . . . . . . . . .

      //test of T-functions
      //         printf("\n");
      //         printf(":::%g ",Fcts.T1(1.0));
      //         printf(":::%g ",Fcts.T2(1.0));
      //         printf(":::%g ",Fcts.T3(1.0));
      //         printf(":::%g ",Fcts.T4(1.0));
      //         printf(":::%g ",Fcts.T5(1.0));
      //         printf(":::%g ",Fcts.T6(1.0));
      //         printf(":::%g ",Fcts.T7(1.0));
      //         printf("\n");


    }
  }
  //...............................................

}


//*******************************************************
//*******************************************************
//CLASS-MEMBER-FUNCTIONS

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
//class Stagetable:

//-------------------------------------------------------
// constructor:
Stagetable::Stagetable(int userk) {
  k = userk;
  State = new int[userk+1];
  Decision = new int[userk+1];
  Cumreturn = new double[userk+1];
}

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
//class Strategy:
```

```
//-----------------------------------------------------------
// constructor:
Strategy::Strategy(int userk) {
  k = userk;
  Decisions = new int[userk+1];
  Stagereturns = new double[userk+1];
}

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
//class Fctclass:

//-----------------------------------------------------------
//constructor:
Fctclass::Fctclass(double* programinput) {
  //userd = 0 or 1
  N = int(programinput[0]);
  k = int(programinput[1]);

  p = new double[int(programinput[1])+1];

  probfct1(programinput[2]);

  parameters = new double[3];
  parameters[0]=1.0;
  parameters[1]=100.0;
  parameters[2]=programinput[3];

}

//-----------------------------------------------------------
// probfct1: generates female's survival
//           probabilities. Exponential decline.
void Fctclass::probfct1(double pee) {
  int i;
  for(i=1; i<=k; ++i) {
    p[i]=pow(pee,i);
  }
// to indicate end of p set last element to -1
// (probabilities are positive)
  p[k+1]=-1.0;
}

//-----------------------------------------------------------
// probfct2: generates female's survival
//           probabilities. Linear decline.
void Fctclass::probfct2(void) {
```

```
  int i;
  double A,B;
  A=1.0; B=150;  // for biggest poss fem N=100
  for(i=1; i<=k; ++i) {
    p[i]= (A/(B-1))*(B - double(i));
  }
// to indicate end of p set last element to -1
// (probabilities are positive)
  p[k+1]=-1.0;
}


    //.....................................
    //.....................................
    // For the moment have to ensure that
    // stagereturn takes correct clutchreturn.
    // Ask Smaug for better solution.
//---------------------------------------------------------
//clutchoffspring:  independent of probabilities,
//                  to be called using pointer.
double Fctclass::clutchoffspring(double decision) {
  double x = decision;
  double b = parameters[0];
  double v = parameters[1];
  double sigma = parameters[2];
  return double( x*b*pow((1-x/v),sigma) );
}

//---------------------------------------------------------
//double Fctclass::clutchgain(double decision) {
  double b = parameters[0];
  double v = parameters[1];
  double sigma = parameters[2];

  return (decision*b*pow((1-decision/v),sigma)*T1(decision));
}



//---------------------------------------------------------
//resourcetransf
double Fctclass::resourcetransf(double decision) {
return T1(decision);
}

//---------------------------------------------------------
```

```
//stageoffspring: depends on probabilities
double Fctclass::stageoffspring(int stage, int decision) {
  return ( p[stage]*clutchoffspring(double(decision))    );

}


//-----------------------------------------------------------
//stagereturn: depends on probabilities
double Fctclass::stagereturn(int stage, int decision) {
  return ( p[stage]*clutchoffspring(double(decision))
          * resourcetransf(double(decision)) );
}
    //.....................................
    //.....................................

//-----------------------------------------------------------
//resultingstate: computes resulting state in next
//                Stagetable from current state and
//                corresponding optimal decision in
//                current Stagetable.
int Fctclass::resultingstate(int state, int decision) {
  return (state - decision);
}


//-----------------------------------------------------------
//getindex: computes index at which state returned by
//          resultingstate is stores in next Stagetable.
int Fctclass::getindex(int lambda) {
  return lambda;
}


//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
//class Toolclass:

//-----------------------------------------------------------
//constructor:
Toolclass::Toolclass(int userk, int Lack_from_main,
     Fctclass& Fcts_from_main) {
  //userd = 0 or 1
  k = userk;
  Lack = Lack_from_main;
  Stagetables = new Stagetable[userk+1](userk+1);
```

```
  pFcts = &Fcts_from_main;
  pclutchoffspring = &Fctclass::clutchoffspring;
  presourcetransf = &Fctclass::resourcetransf;
  pstageoffspring = &Fctclass::stageoffspring;
  pstagereturn = &Fctclass::stagereturn;
  presultingstate =  &Fctclass::resultingstate;
  pgetindex = &Fctclass::getindex;
}

//----------------------------------------------------------
//laststage:  for last stage and for a particular
//            state this computes optdecision and
//            optcumreturn.
void Toolclass::laststage(int state) {
  optdecision = 0; optcumreturn = 0;
  optdecision = min(state,Lack);
  optcumreturn = (pFcts->*pstagereturn)(k,optdecision);
}

//----------------------------------------------------------
//findoptdecision: for a particular stage and state
//                 this computes optdecision and
//                 optcumreturn.
void Toolclass::findoptdecision(int stage, int state) {
  int decision;
  int lambda, index;
  double newcumreturn;
  int maxdecision = min(state,Lack);
  optdecision = 0;
  optcumreturn = 0;

  for (decision = 0; decision <= maxdecision;
                                      ++decision) {
    lambda = (pFcts->*presultingstate)(state,decision);
    index = (pFcts->*pgetindex)(lambda);
    newcumreturn = Stagetables[stage+1].Cumreturn[index]
      +  (pFcts->*pstagereturn)(stage,decision) ;
    if (optcumreturn < newcumreturn) {
      optcumreturn = newcumreturn;
      optdecision = decision;
    }
  }
}

//----------------------------------------------------------
//printStagetable: prints out Stagetable.
```

```
void Toolclass::printStagetable(int which) {
  int i;
  printf("\nstage: %d\n", which);
  for (i=0; i<=k; ++i) {
    printf("state, decision, cumreturn :::  "
           "%d ,  %d ,  %g\n" ,
   Stagetables[which].State[i],
   Stagetables[which].Decision[i],
   Stagetables[which].Cumreturn[i]);
  }
}

//--------------------------------------------------------
void Toolclass::optimise(void) {
  int stage,  state;
  //.............................................
  // Last stage, ie. first stage in optimisation
  // process, has to be done seperately:
  for (state = 0;  state <= k; ++state) {
    laststage(state);
    Stagetables[k].State[state] = state;
    Stagetables[k].Decision[state] = optdecision;
    Stagetables[k].Cumreturn[state] = optcumreturn;
  }
  // printStagetable(k);
  //.............................................
  // Stages k-1 down to 2:
  for (stage = k-1; stage >= 2; --stage) {
    for ( state = 0; state <= k; ++state) {
      findoptdecision(stage, state);
      Stagetables[stage].State[state]=state;
      Stagetables[stage].Decision[state]=optdecision;
      Stagetables[stage].Cumreturn[state]=optcumreturn;
    }
    // printStagetable(stage);
  }
  //.............................................
  // Stage 1 (ie. last stage in process):
  findoptdecision(1,k);
}


//--------------------------------------------------------
//makeStrategy: fills Optstrategy with decisions and corresponding
//              resource returns.
```

```
void Toolclass::makeStrategy(Strategy* pOptstrategy) {
  int i;
  int state=k;
  int decision;

  //.............................................
  //this puts decision and resources gained by clutch into Optstrategy

  //. . . . . . . . . . . . . . . . . . . . . . . . .
  //first clutch
  decision = pOptstrategy->Decisions[1] = optdecision;
  pOptstrategy->Stagereturns[1] =
    (pFcts->*pstagereturn)(1,decision);

  //. . . . . . . . . . . . . . . . . . . . . . . . .
  //other clutches
  for (i=2; i<=k; ++i) {
    state = (pFcts->*presultingstate)(state,decision);
    decision =
      Stagetables[i].Decision[(pFcts->*pgetindex)(state)];
    pOptstrategy->Decisions[i] = decision;
    pOptstrategy->Stagereturns[i] =
      (pFcts->*pstagereturn)(i,decision);
  }

}

//-----------------------------------------------------------
//printStrategy: prints Optstrategy.
void Toolclass::printStrategy(Strategy* pOptstrategy) {
  int i,j,counter,columns;
  int decision;
  int limit = 34;

  //.....................................................
  //this prints Optstrategy
            //  (pFcts->*presourcetransf)(double(decision))  );

  //. . . . . . . . . . . . . . . . . . . . . . . . .
  //first clutch
  printf("&%d ", optdecision );

  //. . . . . . . . . . . . . . . . . . . . . . . . .
  //other clutches
  columns = 2;
  for (i=2; i<=k; ++i) {
```

```
    decision = pOptstrategy->Decisions[i];

    if (decision != 0) {
      if (columns==limit+1) {
printf(" \\\\ \\hline \n");
printf(" && ");
columns = 1;
      }
      printf("&%d ", decision );
    }


    else {
      counter = 1;

      while ( (decision == 0) && (i<k) ) {
++i; decision = pOptstrategy->Decisions[i];
if (decision == 0) counter += 1;
else --i;
      }

      if (columns==limit+1) {
printf(" \\\\ \\hline \n");
printf(" && ");
columns = 1;
      }
      printf("&\\underline{%d} ", counter);
    }

    ++columns;
  }


  if (k<limit) {
    printf(" ");
    for (j=1; j<= (limit-columns)+1; ++j) {
      printf("&");
    }
  }

  if (k>limit) {
    printf(" ");
    for (j=1; j<= (limit-columns)+1; ++j) {
      printf("&");
    }
  }
```

```
  printf(" \\\\ \\hline \n");
  //. . . . . . . . . . . . . . . . . . . . . . . . .

}



//----------------------------------------------------------
//printinfo: prints
//           total invested resources,
//           expected number of survivours,
//           average resources gained by offspring,
//           number of clutches
//           total resources.
void Toolclass::printinfo(Strategy* pOptstrategy) {
  int i;
  int totalres;
  double survivours;
  double clutches;
  double resources;
  double avclutch;
  double percentavclutch;
  double sumofsquarediffs;
  double spread;

  //calculates total resources invested by female
  totalres=0;
  for (i=1; i<=k; ++i) {
    totalres +=  pOptstrategy->Decisions[i];
  }

  //calculates expected number of survivours
  survivours=0;
  for (i=1; i<=k; ++i) {
    survivours +=  (pFcts->*pstageoffspring)
      (i,pOptstrategy->Decisions[i]) ;
  }

  //calculates number of non-zero clutches
  clutches=0.0;
  for (i=1; i<=k; ++i) {
    if (pOptstrategy->Decisions[i] != 0) clutches += 1.0;
  }

  //calculates total resources
```

```
  resources=0;
  for (i=1; i<=k; ++i) {
    resources +=
      (pFcts->*pstageoffspring)(i,pOptstrategy->Decisions[i])*
      (pFcts->*presourcetransf)(double(pOptstrategy->Decisions[i]));
  }

  //calculates average clutch as percentage of Lack clutch
  avclutch = double(totalres)/clutches;
  percentavclutch = (100.0/double(Lack))*avclutch;

  //calculates spread as percentage of Lack clutch
  //first calculate sum of (avclutch-clutch size)^2:
  sumofsquarediffs=0;
  for (i=1; i<=k; ++i) {
    if (pOptstrategy->Decisions[i] != 0)
      sumofsquarediffs += pow(avclutch - double(pOptstrategy->Decisions[i]),2);
  }

  //now calculate (100/Lack)*sqrt((1/k)*sumofsquarediffs):
  spread = (100.0/double(Lack))*sqrt( (1.0/double(k))*sumofsquarediffs );

  //this prints information for Latex table
  printf("&%g &%g &%g &%g &%g \\\\ \\hline \n"
 , survivours, resources, clutches
 , percentavclutch, spread);

}


//*********************************************************
//FUNCTIONS

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// functions that handle user input:

//-------------------------------------------------------
//readfloat: reads in user input and returns it as
//          a float.
double readfloat(void) {
  char c;
  char v[20];
  int i=0;
  while ((c=getchar()) != '\n') {
    v[i]=c;
    ++i;
```

```
  }
  v[i] = '\0';
  return atof(v);
}

//----------------------------------------------------------
// strisnumber: returns 1 if string represents a number,
// 0 otherwise.
int strisnumber(char* s) {
  int i;
  int tmp=1;
  for(i=0; s[i] != '\0'; ++i) {
    if (!isdigit(s[i])) tmp=0;
  }
  return tmp;
}

//----------------------------------------------------------
// inputhandler_Nkptd:

//  * reads in  user input
//  * makes it availble to the program by returning
//    input as array of ints and by calling functions
//    which allows to change parameters of term function
//  * returns k, p, t (int this order) if given by user,
//    otherwise value corresponding to option which was
//    not given will be zero

// Program is run by typing in executable_name followed
// by the options:


// -N eggload                 integer
// -k number_of_stages
// -p probability_function    digit 1, 2, 3, ...
//                            (Look up number of
//                            possible choices in
//                            program.)
// -t term_function           digit 1, 2, 3, ...
//                            (Look up number of
//                            possible choices in
//                            program.)
// -d                         will call function
//                            that changes default
//                            parameters given to
//                            term function
```

```
// Note:
// -N is compulsory, -k,  -p, -t and -d are optional.
// If -k, -p and -t options are not given program will
// choose default functions.



void inputhandler_Nkptd(int* userinput_Nkptd,
int argc,char** argv) {
 int i;
 int N=0, k=0, p=0, t=0, d=0;
  char pN[] = "-N", pk[] = "-k", pp[] = "-p"
     ,  pt[] = "-t", pd[] = "-d";

  //...........................................
  if ((argc <= 1) || (argv[1][0] != '-'))
    printf("Error: first option wrong!\n");

  else {
    for(i=1; (argv[i] != '\0'); ++i) {
      if (strcmp(argv[i],pN)==0) {
  if (strisnumber(argv[++i])) N=atoi(argv[i]);
  else
  printf("Error: no digit follows option -N!\n");
      }
      else if (strcmp(argv[i],pk)==0) {
  if (strisnumber(argv[++i])) k=atoi(argv[i]);
  else
  printf("Error: no digit follows option -k!\n");
      }
      else if (strcmp(argv[i],pp)==0) {
  if (strisnumber(argv[++i])) p=atoi(argv[i]);
  else
  printf("Error: no digit follows option -p!\n");
      }
      else if (strcmp(argv[i],pt)==0) {
  if (strisnumber(argv[++i])) t=atoi(argv[i]);
  else
  printf("Error: no digit follows option -t!\n");
      }
      else if (strcmp(argv[i],pd)==0) {
d=1;
      }
      else {
printf("current argv: %s\n", argv[i]);
```

```
printf("Error: wrong kind of option given!\n");
      }
    }
  }
  //.............................................

  // printf("\n\n");
  userinput_Nkptd[0] = N;
  // if (N==0) printf("zero eggload?\n");
  userinput_Nkptd[1] = k;
  // if (k==0) printf("zero stages?\n");
  userinput_Nkptd[2] = p;
  // if (p==0) printf("No probability function chosen!\n");
  userinput_Nkptd[3] = t;
  // if (t==0) printf("No term function chosen!\n");
  userinput_Nkptd[4] = d;
  // if (d==0) printf("Not opted for change of "
  //                  "parameters!\n");

//   printf("user input: N = %d,  k = %d,  "
//                        "p = %d,  t = %d,  d = %d\n"
//          , userinput_Nkptd[0]
//          , userinput_Nkptd[1]
//          , userinput_Nkptd[2]
//          , userinput_Nkptd[3]
//   , userinput_Nkptd[4]   );


}



//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
//non-member-functions for dynamic programming process:
int Lackfct(Fctclass& Fcts, Fctclass_PFdd pf, double v) {
  int i;
  int opt=0;
  double fopt=0;
  for (i=0; i<= (int) v; ++i) {
    if ( (Fcts.*pf)(double(i)) > fopt ) {
      fopt = (Fcts.*pf)(double(i));
      opt = i;
    }
  }
  return opt;
```

```
}


//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
//Other functions:

//----------------------------------------------------------
int min(int a, int b) {
  return (a <= b) ? a : b;
}

//----------------------------------------------------------
//closestint returns closest integer.
int closestint(double x)  {
int trunc = int(x);
if (x>=0) {
  if   (x<trunc+0.5)  return trunc;
  else                return (trunc+1);
}
else{
  if   (x>trunc-0.5)  return trunc;
  else                return (trunc-1);
}
}

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
//These functions belong to class Fctclass!
//Third factor functions: T1,T2,T3,T4,T4,T5,T6,T7.

//----------------------------------------------------------
// T1: linear decline, steep.
double Fctclass::T1(double x) {
  double v = parameters[1];
  double value;

  if ( (x>=0.0) && (x<1.0) ) {
    value = 0.0;
    return value;
  }
  if ( (x>=1.0) && (x<=v) ) {
    value = -(100.0-33.0)/(v-1)*(x-1.0)+100.0;
    return value;
  }
  if (x>v)  {
    value = 0;
```

```
    return value;
  }
  else {
    printf("Neg val assigned to resourcetransf!\n");
    return 0;
  }
}


//------------------------------------------------------------
// T2: linear decline, less steep.
double Fctclass::T2(double x) {
  double v = parameters[1];
  double value;

  if ( (x>=0.0) && (x<1.0) ) {
    value = 0.0;
    return value;
  }
  if ( (x>=1.0) && (x<=v) ) {
    value = -(80.0-50.0)/(v-1)*(x-1.0)+80.0;
    return value;
  }
  if (x>v)  {
    value = 0;
    return value;
  }
  else {
    printf("Neg val assigned to resourcetransf!\n");
    return 0;
  }
}


//------------------------------------------------------------
// T3: exponential decline.
double Fctclass::T3(double x) {
  double v = parameters[1];
  double value;

  if ( (x>=0.0) && (x<1.0) ) {
    value = 0.0;
    return value;
  }
  if ( (x>=1.0) && (x<=v) ) {
```

```
    value = 100.0*exp(-0.07*(x-1.0));
    return value;
  }
  if (x>v)  {
    value = 0;
    return value;
  }
  else {
    printf("Neg val assigned to resourcetransf!\n");
    return 0;
  }
}


//------------------------------------------------------------
// T4: logarithmic decline.
double Fctclass::T4(double x) {
  double v = parameters[1];
  double value;

  if ( (x>=0.0) && (x<1.0) ) {
    value = 0.0;
    return value;
  }
  if ( (x>=1.0) && (x<=v) ) {
    value = 100.0/log(100.0) * log(-(x-101.0));
    return value;
  }
  if (x>v)  {
    value = 0;
    return value;
  }
  else {
    printf("Neg val assigned to resourcetransf!\n");
    return 0;
  }
}


//------------------------------------------------------------
// T5: sigmoid.
double Fctclass::T5(double x) {
  double v = parameters[1];
```

```
  double beta = 0.1;
  double xco = 50.0;
  double yco = 66.67;

  double value;

  if ( (x>=0.0) && (x<1.0) ) {
    value = 0.0;
    return value;
  }
  if ( (x>=1.0) && (x<=v) ) {
    value = (-1.0)* ((yco-100.0)/tanh(beta*(1.0-xco))) * tanh(beta*(x-xco))
            + yco;
    return value;
  }
  if (x>v)  {
    value = 0;
    return value;
  }
  else {
    printf("Neg val assigned to resourcetransf!\n");
    return 0;
  }
}


//---------------------------------------------------------
// T6: Allee-effect, parabola.
double Fctclass::T6(double x) {
  double v = parameters[1];
  double value;

  if ( (x>=0.0) && (x<1.0) ) {
    value = 0.0;
    return value;
  }
  if ( (x>=1.0) && (x<=v) ) {
    value =
    -19.0/792.0*pow(x,2) + 1319.0/792.0*x
    + 19.0/792.0*10000.0 - 1319.0/792.0*100.0;
    //+ 14524.8/198.0;
    return value;
  }
  if (x>v)  {
    value = 0;
```

```
    return value;
  }
  else {
    printf("Neg val assigned to resourcetransf!\n");
    return 0;
  }
}

//-------------------------------------------------------
// T7: Allee-effect, bell-curve.
double Fctclass::T7(double x) {
  double v = parameters[1];
  double value;

  if ( (x>=0.0) && (x<1.0) ) {
    value = 0.0;
    return value;
  }
  if ( (x>=1.0) && (x<=v) ) {
    value =
    (-0.005513202316*pow(x,2) -0.6441830438*x
    + 0.005513202316*10000.0 +0.6441830438*100.0 + 1.0*pow(10.0,-14))*
    //+119.55032755) *
    exp(-(1.0/(12.5*v))*pow(x-v/4.0,2));
    return value;
  }
  if (x>v)  {
    value = 0;
    return value;
  }
  else {
    printf("Neg val assigned to resourcetransf!\n");
    return 0;
  }
}


//*******************************************************
//REMARKS

// Inputoptions to be given to the program when running it:
//      -N egg load
//      -k number of stages
```

Bibliography

[Allee *et al.* 49] Allee, W. C.; Emerson, A. E.; Park, O.; Park, T.; Schmidt, K. P. 1949. Principles of Animal Ecology. Saunders, Philadelphia.

[Barr *et al.*] Barr, A. R.; Fujioka, K. K.; Wilmot, T. R.; Cope, S. E. 1986. Seasonal variation in number of eggs laid by *Culiseta incidens* (Diptera: Culicidae). *Journal of Medical Entomology* **23**:178-181.

[Begon/Parker 86] Begon, M.; Parker, G. A. 1986. Should egg size and clutch size decrease with age? *Oikos* **47**:293-302.

[Bellman 57] Bellman, R. E. 1957. On the formulation of dynamic-programming problems. Rand Corporation, Santa Monica, CA.

[Charnov 76] Charnov, E. L. 1976. Optimal foraging, the marginal value theorem. *Theor. Popul. Biol.***9**:129-136.

[Charnov/Krebs 74] Charnov, E. L.; Krebs, J. R. 1974. On clutch size and fitness. *Ibis* **116**:217-219.

[Charnov/Skinner 85] Charnov, E. L.; Skinner, W. S. 1985. Complementary approaches to the understanding of parasitoid oviposition decisions. *Environ. Entomol.* **14**:383-391.

[Charnov/Skinner 88] Charnov, E. L.; Skinner, W. S. 1988. Clutch size in parasitoids: the egg production rate as a constraint. *Evol. Ecol.* **2**:167-174.

[Green 89] Green, G. L. M. 1989. An ecological study of *Trichoprosopon Digitatum* (Diptera: culicidae). University of Dundee, PhD. thesis, 1989.

[Godfray 86] Godfray, H. C. J. 1986. The evolution of clutch size in parasitic wasps. *Am. Nat.* **129**:221-234.

[Godfray 87]	Godfray, H. C. J. 1987. The evolution of clutch size in invertebrates. In *Oxford Surveys of Evolutionary Biology*, ed. P. H. Harvey, L. Partridge, **4**:117-154. Oxford Univ. Press, Oxford.
[Godfray/Ives 88]	Godfray H.C.J.; Ives, A. R. 1988. Stochasticity in invertebrate clutch-size models. *Theor. Popul. Biol.***33**:79-101.
[Godfray *et al.* 91]	Godfray, H. C. J.; Partridge, L.; Harvey, P. H. 1991. Clutch size. *Annu. Rev. Ecol.* **22**:409-429.
[Godfray 95]	Godfray, H. C. J. 1995. Evolutionary theory of parent-offspring conflict. *Nature* **376**:133-138.
[Hassell 78]	Hassell, M.P. 1978. The dynamics of arthropod predator-prey systems. Princeton University Press, Princeton, NJ.
[Hubbard 72]	Hubbard, S. F. 1972. The natural mortality of the larval stages of *Pieris brassicae* (Lep.). M.Sc. thesis, University of London.
[Hubbard 77]	Hubbard, S. F. 1977. Studies on the natural control of *Pieris brassicae* (L.) with particular reference to parasitism by *Apanteles glomerata* (L.). Ph.D. thesis, University of Oxford.
[Ives 89]	Ives, A. R. 1989. The optimal clutch size of insects when many females oviposit per patch. *Am. Nat.* **133**: 671-687.
[Iwasa *et al.* 84]	Iwasa, Y.; Suzuki, Y.; Matsuda, H. 1984. Theory of oviposition strategy of parasitoids. I. Effect of mortality and limited egg number. *Theor. Popul. Biol.* **26**:205-227.
[Krebs/McCleery 84]	Krebs, J. R.; McCleery, R. H. 1984. Optimization in behavioural ecology. In *Behavioural Ecology, an Evolutionary Approach*, ed. J. R. Krebs, N. B. Davies, 2nd edn. Blackwell Scientific Publications, Oxford.
[Kristensen 94]	Kristensen, C. O. 1994. Investigations on the natural mortality of eggs and larvae of the large white Pieris brassicae (L.) (Lep., Pieridae). *J. Appl. Entomol.* **117**:92-98.
[Lack 47]	Lack, D. 1947. The significance of clutch size. *Ibis* **89**:309-352.
[Le Masurier 1994]	Le Masurier, D. D. 1994. Costs and benefits of egg clustering in *Pieris brassicae*. *J. Anim. Ecol.* **63**:677-685.

[Mangel 87] Mangel, M. 1987. Oviposition site selection and clutch size in insects. *J. Math. Biol.* **25**:1-22.

[Mangel/Clark 88] Mangel, M.; Clark, C. W. 1988. Dynamic modeling in behavioural ecology. Princeton University Press, Princeton, NJ.

[Mangel 89] Mangel, M. 1989. An evolutionary interpretation of the "motivation to oviposit". *J. evol. Biol.* **2**:157-172.

[Pianka/Parker 75] Pianka, E. R.; Parker, W. S. 1975. Age-specific reproductive tactics. *Am. Nat.* **109**:453-464.

[Parker/Begon 86] Parker, G. A.; Begon, M. 1986. Optimal egg size and clutch size: effects of environment and maternal phenotype. *Am. Nat.* **128**:523-592.

[Parker/Courtney 84] Parker, G. A.; Courtney, S. P. 1984. Models of clutch size in insect oviposition.. *Theor. Popul. Biol.* **26**:27-48.

[Parker/Maynard Smith 90]
Parker, G.A.; Smith, Maynard 1990. Optimality theory in evolutionary biology. *Nature* **348**:27-33.

[Rothschild/Schoonhoven 77]
Rothschild, Miriam; Schoonhoven, L. M. 1977. Assessment of egg load by *Pieris brassicae* (Lepidoptera: Pieridae). *Nature* **266**:352-355.

[Skinner 85] Skinner, S. W. 1985. Clutch size as an optimal foraging problem for insects. *Behav. Ecol. Sociobiol.* **17**:231-238.

[Stephens/Krebs 86] Stephens, D. W.; Krebs, J. R. 1986. Foraging theory. Princeton University Press, Princeton, NJ.

[Suzuki/Iwasa 80] Suzuki, Y.; Iwasa, Y. 1980. A sex ratio theory of gregarious parasitoids. *Res. Popul. Ecol.* **22**:366-382.

[Tammaru *et al.* 96] Tammaru, T.; Kaitaniemi, P.; Ruhomäki, K. 1996. Realized fecundity in *Epirrita autumnata* (Lepidoptera: Geometridae): relation to body size and consequences to population dynamics. *Oikos* **77**:407-416.

[Waage/Godfray 85] Waage, J. K.; Godfray, H. C. J. 1985. Reproductive strategies and population ecology of insect parasitoids. In *Behavioural Ecology*, ed. R. M. Sibly, R. H. Smith, pp. 449-470, Blackwell Scientific Publications, Oxford..

All plots and computational analyses were performed with MapleV.