Evolution of Reaction Norm of Reproductive Diapause in *Callosobruchus maculatus*

A Dissertation Submitted to
the Graduate School of Life and Environmental Sciences,
the University of Tsukuba
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Science.
(Doctoral Program in Integrative Environmental Sciences)

Yukari SUZUKI

CONTENTS

Abstract

Phenotypic plasticity of reproductive diapause is one of strategies to adjust the reproductive schedule to dynamics of resource for oviposition. The change of resource dynamics for oviposition would be a strong selection pressure on the shape of reaction norm of reproductive diapause. The evolution of reproductive diapause in stored-product insects has been considered as the adaptation to resource dynamics from comparative and empirical studies among closely related species. However, this scenario remains to be verified by theoretical models or simulations.

To consider the evolution of reaction norm of reproductive diapause, I focused on *Callosobruchus maculatus* (F.) (Coleoptera: Bruchidae), which exhibits phenotypic plasticity of reproductive diapause induced by temperature increase. I modeled reaction norm by a sigmoid function, and constructed a configuration individual-based model of *C. maculatus*. I constructed simulation models of evolution of reaction norm under different resource dynamics (the field and the storage), and analysed optimal reaction norms in each resource dynamics. From simulation results, I confirmed that plastic reproductive diapause was adaptive in the field condition where temperature and the amount of resource change seasonally. Contrarily, in the storage where the amount of resource is constant, the optimal genotype showed implasticity with short reproductive diapause and high fecundity. However, in heterogeneous condition consisted of both field and storage resources, plastic phenotype became advantageous even in the storage when I incorporated physiological correlation between reproductive diapause and dispersal ability. The phenotypic plasticity was also advantageous when faculty of plasticity (e.g., regulatory genes) and sensitivity to temperature (e.g., sensory genes) were independently genetically controlled. Migration from the storage to the field and gene flow from the field would be influential for the evolution of plastic reproductive diapause, and the adaptation to the storage condition in *C. maculatus*.

keywords: Phenotypic plasticity, Genetic assimilation, Migratory syndrome, Adaptation to the storage

LIST OF TABLES

LIST OF FIGURES

1 Introduction

The environmental change is the primary selection pressure for organisms because the relationship between phenotype and fitness depends on environments. As an environment is often heterogeneous spacially and temporally, organisms may always suffer strong selection pressure for the optimal phenotype. The genotype-environment interaction (phenotypic plasticity) is important in evolutionary biology because phenotypic plasticity sometimes produces phenotypes adapting to each environment without genetic change. The profile of phenotypes produced by a given genotype across environments is known as the norm of reaction (Schmalhausen 1949). Reaction norm with high level of plasticity can be selected for in diverse environments as well as multiple reaction norms with low level of plasticity. The high level of plasticity may lead to generalization through adaptation to diverse environments whereas low level of plasticity may lead to specialization through local adaptation (Pigliucci and Murren 2003).

Reproductive diapause, arrest of sexual development in adult stage, is one example showing high level of phenotypic plasticity. Reproductive diapause can be classified into two types; obligate reproductive diapause, and facultative (plastic) reproductive diapause. Insects showing obligate reproductive diapause enter reproductive diapause in adult stage, and terminate it based on external cues. Contrarily, in plastic reproductive diapause, external cues during sensitive stages (mainly juvenile stages) determine whether reproductive diapause is induced or not in adult stage. The external cues of plastic reproductive diapause (e.g., changes in photoperiod, temperature, humidity, food quality, and population density) generally indicate future environmental and resource conditions. For example, Monarch butterflies (*Danaus plexippus*) enter reproductive diapause by the change of photoperiod whereas grasshoppers (*Anacridium aegyptium*, *Melanoplus devastator*, and *etc.*) enter reproductive diapause by the change of humidity (Tatar and Yin 2001). For *Schistocerca gregaria* and *Locusta migratoria*, preovipositional period is prolonged by high nymphal density (Applebaum and Heifetz 1999).

The acquisition and loss of phenotypic plasticity of reproductive diapause has been studied by comparative method among closely related species in stored-product insects. According to Kiritani (1961), closely related species in stored-product insects could be classified into three types; field type, intermediate type, and storage type (Tables 1.1 and 1.2). Although this classification was mainly based on habitats and digestive ability of dry products, reproductive diapause was also different among these types. Especially in bruchid species, details about reproductive diapause was studied among populations within a species as well as closely related species (Utida 1956; Caswell 1960; Kiritani 1961; Utida 1972; Watanabe 1990; Utida 1998; Zannou et al. 2003). The field type larva feeds on immature seeds in the field. As immature seeds are available for short period, the field type shows long obligate reproductive diapause. The intermediate type inhabits in both field and storage, and larvae can digest raw and dry seeds. It shows plastic reproductive diapause because digestive ability of dry seeds extends the oviposition period when resource is available. The storage type feeds on dry seeds in the storage. It loses reproductive diapause because it can use dry seeds for all seasons. From these comparative studies, constant resource in the storage should be a selection pressure for loss of reproductive diapause. However, the intermediate type exhibits phenotypic plasticity of reproductive diapause even in the storage. If there are some factors to maintain phenotypic plasticity for the intermediate type in the storage, the loss of reproductive diapause would occur when these factors are excluded in the storage.

To consider the acquisition and loss of phenotypic plasticity of reproductive diapause, I focused on the shape of reaction norm of reproductive diapause in the intermediate type. I chose *Callosobruchus maculatus* (F.) (Coleoptera: Bruchidae) as a reference organism of the intermediate type. They infest cultivated legumes in the field during the harvest-time, and stored legumes for all seasons. They show plastic reproductive diapause, and enter reproductive diapause by increase of temperature during sensitive larval stage (Sano 1967; Utida 1998). To simulate the evolution of reaction norms of reproductive diapause, I modeled reaction norms of reproductive diapause by a sigmoid function, and constructed a configuration individual-based model (cIBM).

I simulated three conditions as follows:

1. **Simulation in the field condition**

   To verify that acquisition of phenotypic plasticity of reproductive diapause is adaptive in the field.

2. **Simulation in heterogeneous condition of field and storage**

   To find factors to maintain phenotypic plasticity of reproductive diapause in the storage.

3. **Simulation in the storage condition**

   To verify that loss of phenotypic plasticity of reproductive diapause is adaptive in the storage.

From the first simulation results, I show that acquisition of phenotypic plasticity is adaptive for the intermediate type in the field. In the second simulation, I show that phenotypic plasticity can be maintained in the storage when a physiological correlation between reproductive diapause and flight ability was incorporated. Phenotypic plasticity can be also maintained in the storage when faculty of plasticity (e.g., regulatory genes) and sensitivity to temperature (e.g., sensory genes) were independently genetically controlled. I consider that these physiological and genetic factors prevent the intermediate type from evolution toward the storage type. In the third simulation, I also show that loss of phenotypic plasticity is adaptive for the storage type in the storage condition. Based on these simulation results, I discuss the evolutionary process of loss of reproductive diapause in bruchid species.

2 *Callosobruchus maculatus*

2.1 Life history of flight and flightless forms in *C. maculatus*

C. maculatus has phenotypic plasticity that causes seasonal and density-dependent polyphenism (Utida 1956; Caswell 1960; Utida 1972; Utida 1998; Zannou, Glitho, Huignard, and Monge 2003). Typical two phenotypes are called the flight and the flightless forms by Utida (1954), and the active and the normal forms by Caswell (1960). Flight form is characterized not only by longer reproductive diapause, but also by longer longevity, lower fecundity, and higher flight ability. However, life history of flight form is the same as that of flightless form until emergence (Fig. 2.1). They grow for twenty five days from oviposition as larvae or pupae under the condition of 30 $°C$ and 75% R.H. Sensitive stage to transform the adult to flight form has been reported as the second and the third instar larvae. Sensitive larvae can detect the changes of temperature and water content of the bean, and turn to flight form when temperature increases (Sano 1967). The high metabolic activity for rapid growth in the late-third and the fourth instar larvae increase temperature and water content in the bean.

After emergence, flight form can be distinguished from flightless form by ecological (pre-ovipositional period), morphological (elytral color pattern) and behavioral (flight ability) characteristics. Flightless form can begin to lay eggs on the first- to fourth-day after emergence. Their life time is very short, but fecundity is higher than flight form.

Flight form has been classified into several groups. In the case of classification into three groups, one group of flight form is like flightless form; they have high activity, but they can lay eggs without reproductive quiescence and have short lifetime (Zannou et al. 2003). This group may correspond to melanic flightless form in George and Verma (1994) because they have higher activity than non-melanic flightless form. Hence, this group may be excluded from the classification of flight form. The intermediate group of flight form cannot lay eggs for about two weeks after emergence due to reproductive quiescence. Their reproductive quiescence is terminated without environmental stimuli (Utida 1972; Lextrait et al. 1995). In the extreme flight form, adults enter long reproductive diapause, and survive for three months (George and Verma 1994; Zannou et al. 2003; Appleby and Credland 2001). Flight form in this study refers to the extreme flight form.

2.2 Factors of inducing flight form

Figure 2.2 illustrates factors making a bean weevil flight form. There are three factors inducing flight form; environmental factor, maternal factor, and genetic factor.

Environmental factor inducing flight form is the change of temperature and water content within bean. Such environmental change is determined by complex way; environmental change includes outside environmental change, inside environmental change caused by metabolic activity of larvae in the bean, and inside environmental change of other beans if these beans are stored in mass. Seasonal change of temperature would induce reproductive diapause, and cause seasonal polyphenism of *C. maculatus* in the field condition. Interestingly, density-dependent polyphenism is also caused by increase of temperature. In stored beans, temperature inside the beans increases by larval metabolic activity and affects other beans' temperature. As the population size of the late-third and the fourth instar larvae increases, the percentage of emergence of flight form is affected more by the accumulated change of inside temperature rather than by outside temperature change.

In contrast to the change of temperature, water content inside beans cannot affect other beans' water content under the storage situation. Even with high larval density in a heap of beans, the water content of each bean depends primarily on the larval density in that bean (Sano-Fujii 1984). It would be easy for beans to absorb outside humidity but hard to vaporize water content in beans. As the main environmental factor of density-dependent polymorphism is the change of temperature, I ignored the change of water content in the bean. I describe the changes of outside temperature and inside temperature caused by larvae in “Model”.

The parental age also affects the percentage of flight form among offspring (Sano-Fujii 1979). The parental age is assumed as the factor changing the threshold to transform flight form; the less viable eggs produced by older parents are more susceptible to stimuli which induce the flight form. For simplicity, I did not incorporate the effect of parental age into my simulation to investigate the evolution of physiological function sensing the change of temperature.

The selection and crossing experiments indicate that the percentage of emergence of flight form is heritable (Sano-Fujii 1986). However, the genetic and physiological mechanisms of turning to flight form have not been revealed. So I modeled reaction norm in a sigmoidal function, and genetic systems on a parthenogenetics and a hermaphrodite. Those details are described in “The reaction norms of reproductive diapause” and “Genetic system of reaction norms” in Model.

2.3 *C. maculatus* and closely related species

The evolution of reproductive diapause in bruchid species corresponds to the process of adaptation to the storage condition (Kiritani 1961; Utida 1998). *Bruchus rufimanus Boh.* is a typical field type; it inhabits only in the field, and feed immature *Vicia faba* seeds (Table 1.1). Their phenotype is assumed ancestral because they are uni-voltine, and they can survive only in the field. They show long reproductive diapause that may last on one year. The change of photoperiod, thermo-phase duration, and ingestion of pollen terminate reproductive diapause in adults (Tran and Huignard 1992; Tran et al. 1993).

Bruchidius trifolii and *B. alfierii* were assumed different species at first, but they are ecotypes in the same species. These ecotypes correspond to the field and the intermediate types, respectively. In autumn, they emerge as the field type (*B. trifolii*), and enter long reproductive diapause in the field. They terminate it by ingestion of nectar and pollen of *Trifolium alexandrium* and other species in spring. They lay eggs on immature seeds of *T. alexandrium*, and these eggs emerge as the intermediate type (*B. alfierii*). *B. alfierii* inhabits both field and storage, and can feed on raw and dry seeds. They lay eggs mainly on dry seeds without reproductive diapause.

As an example of the intermediate type, *B. atrolineatus* inhabits in both field and storage. It has been reported that they are bi-voltine, and show plastic reproductive diapause induced by the change of photoperiod and humidity (Monge and Huignard 1991; Lenga et al. 1991; Glitho et al. 1996). The climatic change at host harvest-time increases the percentage of emergence of flightless form. *Acanthoscelides obtectus* also shows plastic reproductive diapause induced by the change of photoperiod and humidity (Hodek et al. 1981). *C. maculatus* has plastic reproductive diapause, and shows seasonal and density-dependent polyphenism. Its reproductive diapause is induced by the change of temperature, humidity, and parental age (Sano 1967; Sano-Fujii 1984; Sano-Fujii 1979).

Several strains in *C. maculatus*, *C. chinensis* and *Zabrotes subfasciatus* that grow in the storage or laboratory are regarded as the storage type. They have no reproductive diapause.

2.4 The conditions leading to the evolution of reproductive diapause

The scenario on the evolution toward the intermediate type (multi-voltinism) from the field type (uni-voltinism) has been proposed by several authors. According to Pichard et al. (1991), multi-voltinism in *A. obtectus* would be a new adaptation that evolved due to stored beans in the storage. However, alternative scenario was proposed in study with phylogeographic methods applied to mitochondrial DNA and nuclear microsatellite molecular markers in *A. obtectus* and *A. obvelatus* (Alvarez et al. 2005). *A. obvelatus* is the sister species of *A. obtectus*, and shows uni-voltinism. If *A. obtectus* diverged from common uni-voltine ancestor of *A. obvelatus* by adaptation to stored beans, the separation between the two species would be after bean domestication. Molecular data indicated that the separation between the two species was before bean domestication. Alvarez et al. (2005) suggested that multi-voltinism is an older trait, adapted to exploit the year-round fruiting of wild beans in relatively aseasonal habitats (South America), and allowed *A. obtectus* to become a pest in bean granaries.

In the case of the genera *Callosobruchus*, it is considered that multi-voltinism evolved by digestive ability of dry and different host beans, and its evolution was escalated by bean domestication. Before domestication, difficulty in lengthening adult life span (e.g., low survival rate during a dry season) is assumed to be one of selection pressures for the acquisition of digestive ability of dry beans (Watanabe 1990). Larval digestive ability of dry beans would enable female adults to lay eggs on mature beans, which would lead to the evolution toward multi-voltinism. After bean domestication, flight form must complete its life cycle only in the field for coexistence of flight and flightless forms in heterogeneous condition of field and storage (Utida 1981). Thus, the oviposition period in the field would be so long that flight form could repeat some generations in the field. Utida (1981) considered that the oviposition period would be extended in the field by the polyphagous habit on, the seasonal succession of seeds of host beans (wild or cultivated) in the field, combined with the joint cultivation of early and late varieties of beans.

The adaptation to the storage would require the preference of oviposition on scattered beans. Morse and Farrell (2005) investigated the evolution of host use in genus *Stator* by molecular phylogenetic analysis. *Stator* could be classified into three oviposition guilds; species ovipositing directly onto the indehiscent pods (guild A), species ovipositing on predispersal beans in dehiscent pods (guild B), and species ovipositing on scattered beans (guild C). From phylogenetic analysis, the transition between ovipositing on pods (guild A) and ovipositing directly seeds (guild B) appeared more constrained than the transition between ovipositing on predispersal (guild B) versus postdispersal seeds (guild C). They suggested that digestive ability of initial tissue (pod or beans) encountered by the first instar larva may have a greater effect on the evolution of oviposition guild than the changes in host searching strategy for oviposition on scattered beans. When once larvae acquired the digestive ability, it would be easy for female adults to shift to oviposit on scattered beans.

In this study, to consider the acquisition of phenotypic plasticity of reproductive diapause, I first ran simulation in the field condition where harvest-time is relatively long. Following simulation in heterogeneous condition of field and storage was conducted to consider the maintenance of phenotypic plasticity of reproductive diapause. I also ran simulation in the storage condition to consider the loss of phenotypic plasticity of reproductive diapause.

3 Model

3.1 The reaction norms of reproductive diapause

To express phenotypic plasticity of reproductive diapause, I modeled the reaction norm in a sigmoidal function (Fig. 3.1). Let $P(dT)$ be the norm of reaction. Input variable, dT, is the change of temperature during the sensitive larval stage. The sigmoidal function $P(dT)$ is represented as follows:

$$P(dT) = \frac{1}{1 + exp(\theta - \omega dT)} \tag{1}$$

where θ is a threshold, and ω is a weight of input variable dT. ω and θ determine the shape of the sigmoidal function. The output value of $P(dT)$ is a liability that the traits become flight form, and it may correspond to the decrease of juvenile hormone (JH). JH is a central hormone controlling flight ability, mating behavior, reproductive diapause, and longevity (for reviews, see Roff 1986; Applebaum and Heifetz 1999; Tatar and Yin 2001).

Blarer and Doebeli (1999) used another type of sigmoidal function for determining the probability of entering diapause. In contrast to Blarer and Doebeli (1999), I used $P(dT)$ for determining the length of reproductive diapause. In my model, each individual has its own $P_i(dT)$, or own ω_i and θ_i parameters.

3.2 Genetic system of reaction norms

3.2.1 Asexually reproducing organism

In asexual reproduction model, *C. maculatus* were modeled as a parthenogenetic population of females. At the beginning of simulation, founder population had ω_i and θ_i that were determined by a random value from a uniform distribution ranging from the minimum to maximum values for that parameter. The minimum and maximum values of ω_i were -5 and 5 whereas those of θ_i were -10 and 10, respectively. The Mersenne Twister method in GNU Scientific Library was used for random number generator in my simulation (Galassi et al. 2003). After reproductive diapause, each female reproduced its copies that had the same ω_i and θ_i. When offspring was reproduced, there were mutational changes against ω_i and θ_i. Each of the two parameters was changed if the uniform random value ranging $[0, 1)$ was smaller than a prespecified mutation rate ($\mu = 0.01$). If mutation occurred, the value of the maternal gene was replaced by a value drawn randomly from a Gaussian distribution centered on the value of the maternal gene with standard deviation of 1% of the difference between the minimum and maximum values for that gene. If a parameter exceeded the limit, it was reset to the nearest limit value, minimum or maximum.

3.2.2 Sexually reproducing organism

At first, I assumed asexual reproducing organisms for simplicity. However, with asexual reproduction, there is no gene recombination of ω_i and θ_i

among individuals by mating. Hence, to consider the effect of recombination of ω_i and θ_i, I also modeled hermaphroditically reproducing organism. I implemented the genetic system as a two-locus multi-allele model. The genotypic value was obtained by the sum of two alleles (i.e., ω_i and θ_i were determined by the sum of ω_{i1} and ω_{i2}, and the sum of θ_{i1} and θ_{i2}, respectively). ω_{ij} and θ_{ij} in the founder population were random values selected from a uniform distribution ranging from the minimum to maximum values of that allele. The minimum and maximum values of each allele were a half in asexual reproduction model because ω_i and θ_i were determined by the sum of the two alleles.

ω_{ij} and θ_{ij} were inherited from their mother and her mating partner. The mutation rate per allele was set to a half of that in asexual reproduction model ($\mu = 0.005$). When mutation occurred, the value of each allele was replaced by a value drawn randomly from a Gaussian distribution centered on the value of the parental allele. The standard deviation used for Gaussian distribution was 1% of the difference between the minimum and maximum values for the sum of that alleles to give the same effect of mutation in asexual reproduction model.

3.3 Temperature

The change of temperature, dT, included external environmental temperature change, and inside temperature change through metabolic activity of influential larvae. The dynamics of external environmental temperature was based on that observed in Niamey region in Niger (BBC Weather; Fig. 3.2). The dynamics of inside heat caused by metabolic activity was approximated by a convex curve that peaks at the 15-th day after oviposition (Fig. 2 in Sano 1967). I approximated the metabolic heat per larva as $0.0055 - 0.00055(age - 15)^2$ (Fig. 3.3). The change of inside temperature was assumed to directly proportional to the number of larvae. Sensitive larvae in the field beans sensed only inside temperature in the bean they infested. In the storage condition, the heat of each bean accumulated and affected other beans' temperature. Thus, sensitive larvae would sense total change of inside heat of all beans.

3.4 Life history traits

I determined that the change of temperature was sensed during sensitive stage ranging from the 10-th to the 12-th day from oviposition (Sano 1967). In my cIBM, sensitive larva could sense the difference between the temperature at the 9-th day and the average temperature from the 10-th to the 12-th day. The difference between two temperatures (dT) was input into the norm of reaction function (P_i). The stage of influential larvae that raises inside temperature by metabolic activity was assumed to be in the period from the 12-th to the 18-th days (Sano 1967). The inside temperature change caused by influential larvae was described above.

I set the age of sexual maturity at $25 + 315P(dT)$ days (Table 3.1). For *C. maculatus*, there seems to be a physiologically negative correlation between the length of reproductive diapause and fecundity, and a positive correlation between the length of reproductive diapause and flight ability (George and Verma 1994; Utida 1998; Zannou et al. 2003). Hence, fecundity was set at $10 + 90(1.0 - P(dT))$, and migration rate from the storage to the field per day was set at $0.003P(dT)$ in simulations of heterogeneous condition of field and storage (Table 3.2). If the output of $P(dT)$ equals to 0.0, a female matures with 100 eggs on the 25-th day. These life history traits are based on those of flightless forms. If the output of $P(dT)$ equals to 1.0, a female matures with 10 eggs on the 340-th day, which was the life history traits of the ancestral univoltinism.

3.5 Translation to configuration individual-based model

For simulating the evolution of reaction norms, I constructed a configuration individual-based model (cIBM). cIBM is very useful for incorporating realistic details of an organism's biology and environment (Fielding 2004; Strand et al. 2002; Hunt and Amdam 2005). I wrote a computer program of cIBM in C^{++}, and ran simulations on a Linux system.

Simulation started with 200 matured adults in the field on the 1st of October in a hypothetical year. All members of the founder population had random ω_i and θ_i values in asexual reproduction model, or random ω_{ij} and θ_{ij} in sexual reproduction model. The phenotype of the founder population was fixed to extreme flight form ($P_i(dT) = 1.0$) regardless of ω_i and θ_i because the phenotype could not be defined without dT. Their residual fecundity and life time were set at ten eggs and ten days, respectively.

In asexual reproduction model, matured adults, which terminated reproductive diapause, could lay eggs without mating. However, in sexual reproduction model, they must find their mating partner from matured adults before oviposition. Matured adults were checked whether they mated every day. If they did not mate, they could not lay eggs.

On each day, if beans existed, matured adults had five chances to lay an egg on a randomly selected bean. Each adult examined surface of a bean whether there were eggs or emergence holes. When they found ten eggs or at least one emergence hole on the bean, they avoided laying eggs on the bean. If they had no egg or no residual life time, they were discarded from the system. Besides, the density-independent death rate was set at 0.006 per day regardless of their age. An individual died if the uniform random value ranging $[0, 1)$ was smaller than 0.006.

Surviving larva whose age was from 9 to 12 days after oviposition assessed temperature change. The larva whose age was from 12 to 18 days changed inside temperature in their beans. It has been reported that *C. maculatus* larvae compete for resource during larval stage (Takano et al. 2001). However, I did not incorporate larval exploitative competition over limited resource in a bean for simplicity. Larvae inside the beans emerged at 25 days old. Their life history traits were determined by $P_i(dT)$. They could not lay eggs until they reached their mature age, or $25 + 315(1.0 - P_i(dT))$. After sexual maturity, they began to lay eggs. The fecundity was also set at $10 + 90P_i(dT)$. They died at the day they finished to lay all eggs, or the 30-th day from their mature age, i.e., $55 + 315(1.0 - P_i(dT))$.

For convenience, I set one year at 360 days, and one month at 30 days. Above procedure continued for 50 years (18000 days). I ran 100 simulations for analyzing optimal reaction norm.

3.6 Statistics of simulation results

In this paper, I call a set of parameters of reaction norm (ω_i, θ_i) the "genotype" of an individual, which determines the strategy responding to temperature change for adapting resource dynamics. However, since ω_i and θ_i are continuous values, it is difficult to analyze a function of reaction norm from ω_i and θ_i directly. For analyzing the function of reaction norm, I classified genotypes into seven types based on following criteria. At first, genotypes were divided into plastic and implastic ones based on a difference between an output value of reaction norm (P_i) with $dT = 0$ and that with $dT = -2$ or $dT = 2$. When either $|P_i(2) - P_i(0)|$ or $|P_i(0) - P_i(-2)|$ was larger than 0.5, I defined that the genotypes showed plasticity against temperature changes (Table 3.2). The other genotypes were classified into implastic one.

Plastic genotypes were further classified into four types according to $P_i(2)$, $P_i(0)$, and $P_i(-2)$. If $P_i(2) - P_i(0) > 0.5$, the genotypes were a

Plastic Flight form induced by Increase of temperature (PFI). PFI was characterized by $P_i(0) < 0.5$, and the increase of $P_i(dT)$ by increase of dT. Hence, PFI was near to flightless form without temperature change, and it became flight form when temperature increased. I considered that PFI could represent reaction norm of reproductive diapause in *C. maculatus.* If $P_i(0) - P_i(-2) > 0.5$, the genotypes were classified as a Plastic flightLess form induced by Decrease of temperature (PLD). For the other plastic genotypes, Plastic flightLess form induced by Increase of temperature (PLI) and Plastic Flight form induced by Decrease of temperature (PFD) were determined by the condition of $P_i(0) - P_i(2) > 0.5$ and $P_i(-2) - P_i(0) > 0.5$, respectively.

I classified implastic genotypes into three types according to $P_i(0)$. When $P_i(0)$ was higher than 0.7, the genotypes were defined as an Implastic flightLess form (IL). When $P_i(0) < 0.3$, the genotypes were an Implastic Flight form (IF), and the other implastic genotypes were an Implastic Intermediate form (II).

At the end of simulation, all reaction norms among adults were classified into seven types based on this classification (Table 3.2, Fig. 3.4). I investigated the dominant type among adults in each simulation. I determined that a type was dominated in a population if the percentage of the type among adults exceeded 70%. The other case was defined as Genotypic Polymorphism (GP). Afterwards, I analyzed more details about reaction norms and population dynamics.

4 Acquisition of phenotypic plasticity of reproductive diapause in the field

4.1 Resource dynamics in the field condition

I assumed that the amount of resource changes seasonally in the field. In simulation of the field condition, 20 beans were provided in the field every day for 180 days (Fig. 4.1(a)). On the 240-th day, all field beans were removed with pre-adults (i.e., eggs, larvae, and pupae) in the bean due to decay of beans in rainy season (Fig. 4.1(c)).

4.2 Simulation of asexual reproduction model in the field condition

4.2.1 The dominant type

In simulation of the field condition, PFI became to be dominant overwhelmingly (Table 4.1). Eighty six cases were resulted in the dominance of PFI. The fixation of PFI was observed in 15 cases. PFI was advantageous because seasonally changing temperature could be a cue for adjusting reproductive diapause against seasonal resource.

4.2.2 Typical reaction norms

Typical reaction norm of PFI could be classified into two types; $(\omega, \theta) \approx (2.5 \sim 5.0, 0.0 \sim 1.0)$ or $(4.0 \sim 5.0,\ 1.0 \sim 2.0)$ (Fig. 4.2). The former (bPFI) was the reaction norm of bi-voltine individuals whereas the latter (tPFI) was the reaction norm of tri-voltine individuals. Among PFI dominant cases, 62 cases were bPFI dominance. The other cases showed mixture of bPFI and tPFI individuals.

4.3 Simulation of sexual reproduction model in the field condition

4.3.1 The dominant type

The dominance of PFI was observed in 69 cases (Table 4.2). The number of PFI dominant cases decreased compared with those in simulation of asexual reproduction model. Instead of PFI dominant cases, the number of GP cases increased from 4 to 22.

4.3.2 Typical reaction norms and genetic variation

In sexual reproduction model, it is hard to count the number of voltinism in each genotype because a genotype reproduces different genotypes by mating. Thus, we call bPFI and tPFI from ω_i and θ_i observed in simulations of asexual reproduction model, not the number of voltinism in simulations of sexual reproduction model. Among PFI dominant cases, 29 cases showed dominance of bPFI (Fig. 4.3). The coexistence of bPFI and tPFI was observed in 26 cases. Twelve cases showed small genetic variation of ω ($\omega \approx 5.0$) but large genetic variation of θ (Fig. 4.4). The large genetic variation of θ was attributed to the genotypic polymorphism of tPFI ($\theta \approx 1.5$), bPLD ($\theta \approx -0.5$), and sometimes uPLD ($\theta \approx -6.0 \sim -4.0$) (Figs. 4.3 and 4.4). The other two cases were resulted in the coexistence of bPFI and PLD.

4.4 Discussion

PFI was advantageous when seasonally changing temperature could be a cue to adjust reproductive diapause to the resource dynamics. PFI laid eggs during the harvest-time, and it entered reproductive diapause during the fallowing and planting seasons.

As sexual reproduction reduced loss of genetic variation, genotypic polyphenism was maintained frequently. Interestingly, genetic variation of θ tended to be larger than genetic variation of ω. The genetic variation of ω was very small, and mean value of ω was high ($\omega \approx 5.0$). It meant that reaction norms represented almost discrete change of traits, but its sensitivity was different among genotypes. During fallowing and planting seasons, genotype who had small θ was advantageous because such genotype could enter long reproductive diapause. Such genotype was disadvantage during harvest-time due to low fecundity. However, if it mated with a genotype who had positive θ, it could leave its alleles because its mating partner had high fecundity. Hence, negative θ persisted despite of its disadvantage in harvest-time.

5 Maintenance of phenotypic plasticity of reproductive diapause in the field and the storage

5.1 Resource dynamics in heterogeneous condition of field and storage

I set the field condition with seasonally changing resource, and the storage condition with constant resource. I assumed that several field beans were harvested and brought into the storage. For the first half of each year, 25 beans were provided in the field every day. Five field beans were selected randomly, and stored in the storage (Fig. 5.1(a)). With harvested beans, pre-adults in the harvested beans could migrate from the field to the storage for 180 days. On the 240-th day, field beans were removed with pre-adults due to decay of beans in rainy season (Fig. 5.1(c)).

In the storage, the number of total beans was always 1800. This number was relatively large for simulation, but it was only a small amount in real world. Huge amount of stored beans kept constant resource availability, which would be a selection pressure for the storage type. To maintain constant resource availability, I set high turn-over rate of stored beans in the storage. Ten beans in stored beans were randomly selected every day, and removed with pre-adults in them (Fig. 5.1). After removing ten beans, five new beans and five field beans were provided in the storage for the first half of each year (Fig. 5.1(a)). For the second half of each year, ten new beans were provided in the storage (Fig. 5.1(b) and (c)).

5.2 Simulation of asexual reproduction model in heterogeneous condition of field and storage

5.2.1 The dominant type

This simulation also showed PFI dominance in the field. Seventy nine cases resulted in the dominance of PFI, and 19 cases resulted in fixation of PFI (Field in Table 5.1). The constant resource availability in the storage caused the dominance of IL, which is a implastic phenotype with short diapause and high fecundity (Storage in Table 5.1). However, PFI could also be advantageous in the storage (29 cases).

5.2.2 Typical reaction norms and population dynamics

PFI was classified into three types; bPFI, tPFI and sPFI (Fig. 5.2). The reaction norms of bPFI and tPFI were also observed in simulation of asexual reproduction model in the field condition (Fig. 4.2), but sPFI ($\theta \approx 4.0$) was observed for the first time. sPFI was observed in both field and storage. IL was mainly observed in the storage (Fig. 5.2).

Among cases of PFI dominance in the field, 30 cases were that PFI was dominant in the field, and IL was dominant in the storage. sPFI was frequently observed in the storage when bPFI was dominant in the field (17 cases). In the other cases of sPFI observed, nine cases showed sPFI fixation in both field and storage.

In the cases where bPFI and sPFI were observed in the field, sPFI adults showed short reproductive diapause under small temperature change, and repeated short life-cycle during winter when bPFI entered reproductive diapause (Fig. 5.3). Some portions of sPFI pre-adults could return to the storage with harvested beans. From May to October, sPFI born in the field could not survive, but several sPFI migrated from the storage could survive in the field (Fig. 5.4). As temperature in the storage was determined by the sum of outside temperature and accumulated inside temperature, temperature in the storage was always higher than that in the field. Larvae of sPFI in harvested beans sensed great increase of temperature in the storage during May. $P_i(dT)$ of such sPFI was sufficiently high ($0.5 \sim 1.0$), which led to high flight ability to the field, and long reproductive diapause. Thus, sPFI migrants could survive until harvest-time in the field, and lay eggs on field beans. The flight ability kept the advantage of phenotypic plasticity of reproductive diapause even in the storage.

Simulation without migration from the storage to the field showed no cases where PFI was dominant (Storage in Table 5.2). If migration was not incorporated, the constant resource availability in the storage led either IL dominance or genotypic polymorphism.

5.3 Simulation of sexual reproduction model in heterogeneous condition of field and storage

5.3.1 The dominant type

Ninety two cases showed dominance of PFI in the field (Field in Table 5.3). Surprisingly, 94 cases showed PFI dominance in the storage (Storage in

Table 5.3). Simulation without migration also showed PFI dominance, though the number of PFI dominant case decreased to 81 and 78 in the field and the storage, respectively (Table 5.4). Gene recombination and gene flow from the field prevented the evolution of specialized type (IL) in the storage.

5.3.2 Typical reaction norms and genetic variation

Among PFI dominant cases in the field, eight cases showed bPFI dominance, and 49 cases showed the coexistence of bPFI and tPFI (Figs. 5.5 and 5.6). The other cases showed the coexistence of PFI and PLD, like simulation of sexual reproduction model in the field (Fig. 4.4).

In contrast to asexual reproduction model, most cases resulted in PFI dominance even in the storage. Most PFI dominant cases in both the field and the storage resulted in the dominance of bPFI and tPFI in the field, and genotypic polymorphism of bPFI, tPFI, sPFI, and nPFI (49 cases). In the storage population, ω was the same as that in the field, but genetic variation of θ was larger than that in the field. These genetic variation produced the reaction norms of bPFI, tPFI, sPFI, and nPFI (Figs. 5.5 and 5.7).

5.4 Discussion

5.4.1 Flight ability

In simulation of asexual reproduction model in heterogeneous condition of field and storage, PFI was advantageous in the field, and IL and PFI were advantageous in the storage. In the storage, the constant resource availability led to the evolution of high fecundity and short life-cycle. However, occasional migration to the field by flight ability maintained phenotypic plasticity even in the storage. sPFI can migrate to the field, and survive until the next harvest-time. In harvest-time, they lay eggs after short reproductive diapause during winter when bPFI entered reproductive diapause in the field.

"Migratory syndrome" is the phenotypic plasticity with a positive correlation between flight ability and reproductive diapause, which is regulated at physiological level by JH titer (Fairbairn 1994; Dingle and Winchell 1997). Based on this physiological correlation, Fairbairn (1994) also assumed positive correlation between flight ability and reproductive diapause in her model. Generally, flight ability could be advantageous in spatially and temporally heterogeneous environment (Roff 1994).

However, the independence between dispersion and diapause sometimes leads to antagonistic relationship between these traits, which seems to be opposed to migratory syndrome. Wiener and Tuljapurkar (1994) assumed no correlation between migration and diapause, and found that higher levels of diapause reduced the levels of migration. When we consider the optimal reaction norm in seasonal and density-dependent polyphenism, it would be important to incorporate the physiological correlation between flight ability and reproductive diapause.

5.4.2 Gene recombination and gene flow

In contrast to simulation of asexual reproduction model, most cases resulted in PFI dominance even in the storage in simulations with sexual reproduction. Asexual reproduction led to the divergence of ω in each condition frequently. However, sexual reproduction led to the fixation of ω in both field and storage. The genetic differentiation of θ between the field and the storage enabled offspring to adapt to resource availability in each condition. If faculty of phenotypic plasticity ω (e.g., regulatory genes) and sensitivity to temperature θ (e.g., sensory genes) controlled by independent loci, gene flow from the field would prevent the loss of phenotypic plasticity in the storage. For adaptation to the storage condition, insensitivity would be easier to occur than the loss of plasticity.

6 Loss of phenotypic plasticity of reproductive diapause in the storage

6.1 Resource dynamics in the storage condition

In the storage, the number of total beans was set to be always 1800. Ten beans in stored beans were randomly selected every day, and removed with pre-adults in them. After removing ten beans, ten new beans were provided in the storage for all seasons (Fig. 6.1).

6.2 Simulations of asexual reproduction model in the storage condition

6.2.1 The dominant type

The constant resource availability led to loss of phenotypic plasticity. All simulations showed IL dominance, and 65 cases resulted in IL fixation (Table 6.1).

6.2.2 Typical reaction norms

The typical reaction norms was almost linear due to relatively high positive θ (≈ 5.0). It could express very short pre-oviposition period regardless of temperature change (Figure 6.2). However, $P(0.0)$ was not equal to zero, which showed pre-ovipositional period for about three days. It would be attributed to slightly higher death rate of pre-adult stage by high turn-over rate in the storage. The higher death rate in pre-adult stage would decrease fitness more than lower fecundity after pre-ovipositional period.

6.3 Simulations of sexual reproduction model in the storage condition

6.3.1 The dominant type

Ninety five cases showed IL dominance, and five cases resulted in GP (genotypic polymorphism) (Table 6.2). In contrast to asexual reproduction model, phenotypic plasticity was maintained at low level in all replicates.

6.3.2 Typical reaction norms and genetic variation

Large genetic variation of ω produced the reaction norms of PFI, IL, and PFD (Figs. 6.3 and 6.4). The genotype which lays eggs with short reproductive diapause under changeable temperature becomes to be advantageous in the storage. Selection pressure for positive θ was strong, but selection pressure for ω was weaker than that of θ. Thus, positive θ was selected, but genetic variation of ω was maintained by mating in sexual reproduction model (Fig. 6.4).

6.4 Discussion

The constant resource availability led to the loss of reproductive diapause in simulations in the storage. When reproduction was asexual, phenotypic plasticity was lost. However, under sexual reproduction, phenotypic plasticity was maintained at low level by large genetic variation of ω.

These simulation results indicated that the first step of adaptation to the storage condition might be acquiring insensitivity to small increase of temperature. As the physiological correlation between flight ability and reproductive diapause, or gene flow from field maintained plasticity in the storage, insensitivity would be easier to acquire than the loss of plasticity (Chapter 5). The insensitivity would be escalated by temperature increase through larval metabolic activity in the storage condition. Such enhanced insensitivity would cause flight inability, and diminish the advantage of phenotypic plasticity. Even if loci of faculty of plasticity and sensitivity are unlinked, the faculty of phenotypic plasticity would be lost in the storage after evolution toward flight inability, and isolation from field population. Afterwards, bruchid species that lost faculty of plastic diapause, would become the storage type adapting to the storage condition.

It has been reported that reproductive diapause is easier to be lost than flight ability (Harrison 1980; Roff 1986). This is very interesting because results of my simulations showed that flight inability would increase the advantage of implastic normal form in the storage. When the reproductive diapause is easier to be lost, phenotypic plasticity of flight ability would keep the advantage of phenotypic plasticity of reproductive diapause. However, if once flight ability is lost, it would lead to the loss of phenotypic plasticity in the storage.

7 Conclusion

When seasonally changing temperature can be a cue to adjust reproductive diapause to the resource dynamics, plastic reproductive diapause induced by increase of temperature was advantageous because such plasticity enabled the weevils to lay eggs during the harvest-time, and to enter reproductive diapause during the fallowing or planting seasons in the field. On the other hand, in the storage, the constant resource availability led to the evolution toward implastic genotype with high fecundity and short life-cycle. However, under heterogeneous condition of field and storage, phenotypic plasticity could be maintained even in the storage.

There are two factors to maintain phenotypic plasticity. One is physiologically positive correlation between reproductive diapause and flight ability, and the other is genetic unlinkage between locus of plasticity and locus of sensitivity. Migration from the storage to the field by flight maintained the advantage of plastic reproductive diapause. Gene flow from the field also maintained plastic reproductive diapause in the storage.

I suggest that the first step of adaptation to the storage condition might be insensitivity to small increase of temperature. As the correlation between flight ability and reproductive diapause, or gene flow from the field maintained plasticity in the storage, insensitivity would be easier to occur than the loss of plasticity. Afterwards, bruchid species that can use constant resource would lose faculty of plastic diapause, and adapt to the storage condition.

The acquisition and loss of phenotypic plasticity of reproductive diapause could be discussed from theoretical approach. Over 40 species of stored-product insects also exhibit diapause in their life cycle (Bell 1994). The evolution of diapause in these insects could be discussed in the light of the evolution of reaction norm of diapause for the adaptation to resource dynamics.

8 Future works

The present model could show the scenario of the evolution of reaction norms reasonably well. However, it might not show the evolution of voltinism precisely. In both cases of bi-voltinism and multi-voltinism, some portions of adults terminated reproductive diapause before harvest-time, and died without oviposition. It would indicate that the mechanism inducing reproductive diapause should be more complex to adapt resource dynamics.

The induction of reproductive diapause is sometimes caused by interaction of several external factors. I used the change of temperature as the external factor inducing reproductive diapause, but water content in the bean and parental age also affected reproductive diapause in *C. maculatus*. These two factors would be minor factors, but they might prolong the length of reproductive diapause so that reproductive diapause is terminated in harvest-time.

Another factor that affects voltinism is the temperature-dependent developmental rate. I set temperature-independent developmental rate because I focused only on plastic reproductive diapause. However, developmental rate of *C. maculatus* is controlled by temperature. Low temperature in December and January changes sensitive age, which might prevent from termination of reproductive diapause before harvest-time. If we require the model to simulate the evolution of voltinism precisely, more complex model must be constructed.

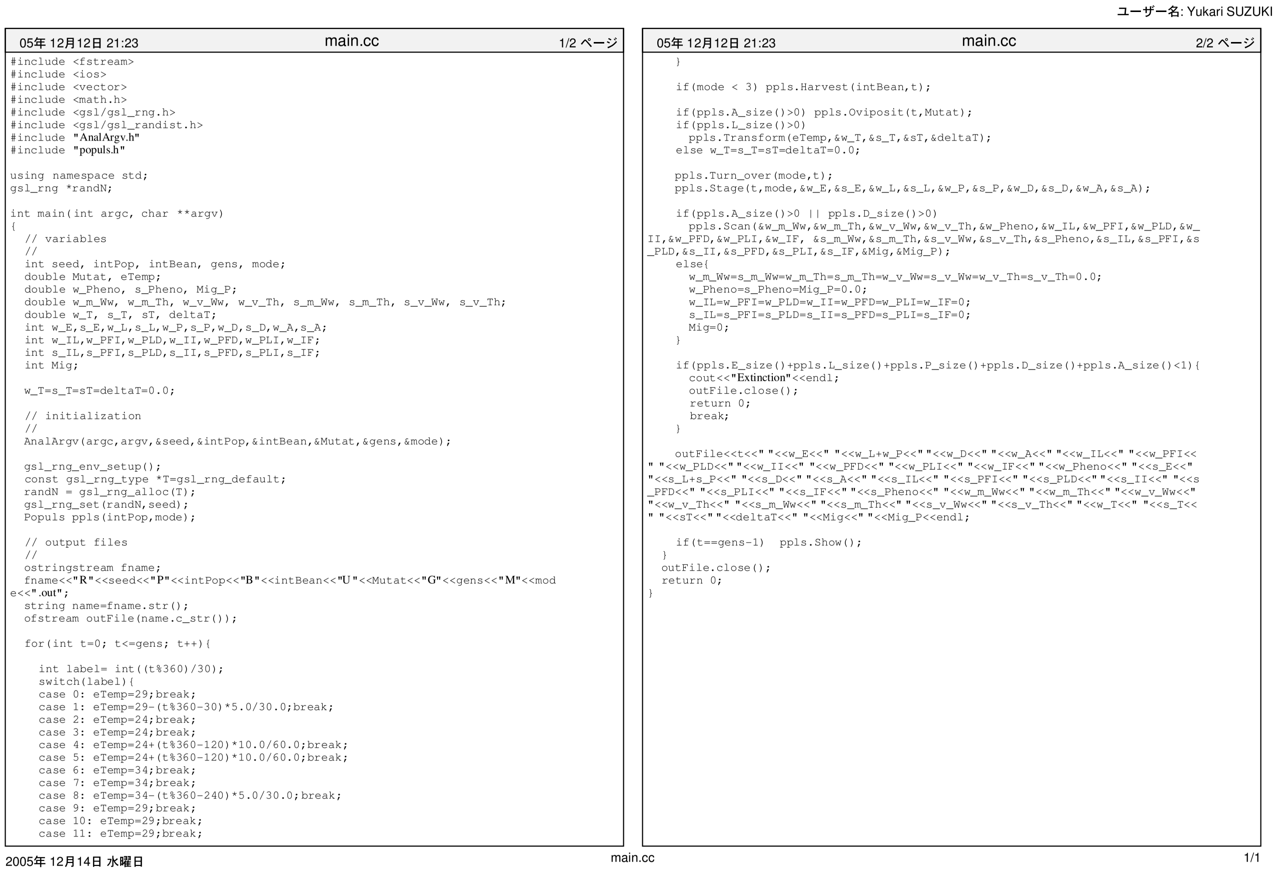
9 Acknowledgements

This study is dedicated to Dr. Shunro Utida, who is the pioneer of experimental population ecology using bruchid species in Japan. I thank Professor Koichi Fujii, Associate Professor Yukihiko Toquenaga, and Assistant Professor Kazuharu Ohashi for their educational guidance. I also thank Professor Mamoru Watanabe and Associate Professor Shigeru Mariko for critical reading and comments.

My model was improved by discussion with my laboratory members in seminar about polyphenism. It was very helpful to discuss about polyphenism of *C. maculatus* with Hiroyuki Mano.

I thank Dr. Kazunori Ohno, who is a Research Associate in Tohoku University, for his encouragement to find novel scientific subjects. Lastly, I especially appeciate my parents' generosity in supporting me spiritually and financially.

Appendix: Programs for asexual reproduction model



ユーザー名: Yukari SUZUKI

05年 12月12日 21:23 main.cc 1/2 ページ

```
#include <fstream>
#include <ios>
#include <vector>
#include <math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
#include "AnalArgv.h"
#include "populs.h"

using namespace std;
gsl_rng *randN;

int main(int argc, char **argv)
{
  // variables
  //
  int seed, intPop, intBean, gens, mode;
  double Mutat, eTemp;
  double w_Pheno, s_Pheno, Mig_P;
  double w_m_Ww, w_m_Th, w_v_Ww, w_v_Th, s_m_Ww, s_m_Th, s_v_Ww, s_v_Th;
  double w_T, s_T, sT, deltaT;
  int w_E,s_E,w_L,s_L,w_P,s_P,w_D,s_D,w_A,s_A;
  int w_IL,w_PFI,w_PLD,w_II,w_PFD,w_PLI,w_IF;
  int s_IL,s_PFI,s_PLD,s_II,s_PFD,s_PLI,s_IF;
  int Mig;

  w_T=s_T=sT=deltaT=0.0;

  // initialization
  //
  AnalArgv(argc,argv,&seed,&intPop,&intBean,&Mutat,&gens,&mode);

  gsl_rng_env_setup();
  const gsl_rng_type *T=gsl_rng_default;
  randN = gsl_rng_alloc(T);
  gsl_rng_set(randN,seed);
  Populs ppls(intPop,mode);

  // output files
  //
  ostringstream fname;
  fname<<"R"<<seed<<"P"<<intPop<<"B"<<intBean<<"U"<<Mutat<<"G"<<gens<<"M"<<mod
e<<".out";
  string name=fname.str();
  ofstream outFile(name.c_str());

  for(int t=0; t<=gens; t++){

    int label= int((t%360)/30);
    switch(label){
    case 0: eTemp=29;break;
    case 1: eTemp=29-(t%360-30)*5.0/30.0;break;
    case 2: eTemp=24;break;
    case 3: eTemp=24;break;
    case 4: eTemp=24+(t%360-120)*10.0/60.0;break;
    case 5: eTemp=24+(t%360-120)*10.0/60.0;break;
    case 6: eTemp=34;break;
    case 7: eTemp=34;break;
    case 8: eTemp=34-(t%360-240)*5.0/30.0;break;
    case 9: eTemp=29;break;
    case 10: eTemp=29;break;
    case 11: eTemp=29;break;
```

05年 12月12日 21:23 main.cc 2/2 ページ

```
    }

    if(mode < 3) ppls.Harvest(intBean,t);

    if(ppls.A_size()>0) ppls.Oviposit(t,Mutat);
    if(ppls.L_size()>0)
      ppls.Transform(eTemp,&w_T,&s_T,&sT,&deltaT);
    else w_T=s_T=sT=deltaT=0.0;

    ppls.Turn_over(mode,t);
    ppls.Stage(t,mode,&w_E,&s_E,&w_L,&s_L,&w_P,&s_P,&w_D,&s_D,&w_A,&s_A);

    if(ppls.A_size()>0 || ppls.D_size()>0)
      ppls.Scan(&w_m_Ww,&w_m_Th,&w_v_Ww,&w_v_Th,&w_Pheno,&w_IL,&w_PFI,&w_PLD,&w_
II,&w_PFD,&w_PLI,&w_IF, &s_m_Ww,&s_m_Th,&s_v_Ww,&s_v_Th,&s_Pheno,&s_IL,&s_PFI,&s
_PLD,&s_II,&s_PFD,&s_PLI,&s_IF,&Mig,&Mig_P);
    else{
      w_m_Ww=s_m_Ww=w_m_Th=s_m_Th=w_v_Ww=s_v_Ww=w_v_Th=s_v_Th=0.0;
      w_Pheno=s_Pheno=Mig_P=0.0;
      w_IL=w_PFI=w_PLD=w_II=w_PFD=w_PLI=w_IF=0;
      s_IL=s_PFI=s_PLD=s_II=s_PFD=s_PLI=s_IF=0;
      Mig=0;
    }

    if(ppls.E_size()+ppls.L_size()+ppls.P_size()+ppls.D_size()+ppls.A_size()<1){
      cout<<"Extinction"<<endl;
      outFile.close();
      return 0;
      break;
    }

    outFile<<t<<" "<<w_E<<" "<<w_L+w_P<<" "<<w_D<<" "<<w_A<<" "<<w_IL<<" "<<w_PFI<<
" "<<w_PLD<<" "<<w_II<<" "<<w_PFD<<" "<<w_PLI<<" "<<w_IF<<" "<<w_Pheno<<" "<<s_E<<"
"<<s_L+s_P<<" "<<s_D<<" "<<s_A<<" "<<s_IL<<" "<<s_PFI<<" "<<s_PLD<<" "<<s_II<<" "<<s
_PFD<<" "<<s_PLI<<" "<<s_IF<<" "<<s_Pheno<<" "<<w_m_Ww<<" "<<w_m_Th<<" "<<w_v_Ww<<"
"<<w_v_Th<<" "<<s_m_Ww<<" "<<s_m_Th<<" "<<s_v_Ww<<" "<<s_v_Th<<" "<<w_T<<" "<<s_T<<
" "<<sT<<" "<<deltaT<<" "<<Mig<<" "<<Mig_P<<endl;

    if(t==gens-1)  ppls.Show();
  }
  outFile.close();
  return 0;
}
```

2005年 12月14日 水曜日 main.cc 1/1

05年 12月12日 21:10 AnalArgv.h 1/1 ページ

```
#ifndef ANALARGV_H
#define ANALARGV_H

#include <iostream>
#include <string>
#include <sstream>

using namespace std;

void AnalArgv(int,char**,int*,int*,int*,double*,int*,int*);

#endif //ANALARGV_H
```

```
#include "AnalArgv.h"

void AnalArgv(int ac, char **av,
              int *randSeed,
              int *numPopul,
              int *numBean,
              double *Mutat,
              int *gens,
              int *mode)
{
  int count=0;

  if(av[1]){
    for(int i=1; i<ac; i++){
      string input(av[i]);
      if(input.find("-rs")==0){
         string input(av[++i]);
         istringstream istr(input);
         istr>>*randSeed;
         count++;
      }
      else if(input.find("-np")==0){
         string input(av[++i]);
         istringstream istr(input);
         istr>>*numPopul;
         count++;
      }
      else if(input.find("-nb")==0){
         string input(av[++i]);
         istringstream istr(input);
         istr>>*numBean;
         count++;
      }
      else if(input.find("-mu")==0){
         string input(av[++i]);
         istringstream istr(input);
         istr>>*Mutat;
         count++;
      }
      else if(input.find("-gn")==0){
         string input(av[++i]);
         istringstream istr(input);
         istr>>*gens;
         count++;
      }
      else if(input.find("-md")==0){
         string input(av[++i]);
         istringstream istr(input);
         istr>>*mode;
         count++;
      }
    }
    if(count<6){
      cout<<endl<<"You should specify a random seed, etc."<<endl<<
         "Just hit ./BW, and you will find the right usage."<<endl<<endl;
      exit(1);
    }
  }
  else{
    cout<<endl<<
      "--------------------------------------------------------------"<<endl<<
      "BW effect                                        "<<endl<<
```

```
      "--------------------------------------------------------------"<<endl<<
      "Specify the following eight parameters in the command line.    "<<endl<<
      "                                                               "<<endl<<
      "./BW -rs random seed -np population size -nb number of beans   "<<endl<<
      "    -mu mutation rate -gn generations                          "<<endl<<
      "    -md condition (0 = field, 1 = field + storage              "<<endl<<
      "              2 = field + storage + migration, 3=storage)"<<endl<<
      "--------------------------------------------------------------"<<endl<<
      endl;
    exit(1);
  }
};
```

```
#ifndef POPULS_H
#define POPULS_H

#include <vector>
#include <algorithm>
#include <fstream>
#include <ios>

#include "popul.h"
#include "bean.h"
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>

using namespace std;
extern gsl_rng *randN;
class Bean;
class Popul;

class Populs{
 private:
  vector<Popul*> eggs,larvae,pupae,diapause,adults;
  vector<Bean*> s_beans,w_beans,waste;

 public:
  Populs(int,int);
  ~Populs();
  void Oviposit(int,double);
  void Transform(double,double*,double*,double*,double*);
  void Stage(int,int,int*,int*,int*,int*,int*,int*,int*,int*,int*,int*);
  void Shuffle();
  int E_size();
  int L_size();
  int P_size();
  int D_size();
  int A_size();
  int B_size();
  void Scan(double*,double*,double*,double*,double*,int*,int*,int*,int*,int*,int
*,int*,double*,double*,double*,double*,double*,int*,int*,int*,int*,int*,int*,int
*,int*,double*);
  void Show();
  void Show(int);
  void Harvest(int,int);
  void Turn_over(int,int);
  void B_Shuffle();
 };

#endif //POPULS_H
```

```
#include "populs.h"

Populs::Populs(int n, int mode){
  for(int i=0; i<n; i++){
    Popul *p = new Popul(mode);
    adults.push_back(p);
  }
  if(mode>0){
    for(int j=0; j<1800; j++){
      Bean *b = new Bean(1);
      s_beans.push_back(b);
    }
  }
}

Populs::~Populs(){
  for(vector<Popul*>::iterator is=eggs.begin();is!=eggs.end();is++)
    delete *is;
  for(vector<Popul*>::iterator it=larvae.begin();it!=larvae.end();it++)
    delete *it;
  for(vector<Popul*>::iterator iu=pupae.begin();iu!=pupae.end();iu++)
    delete *iu;
  for(vector<Popul*>::iterator iv=diapause.begin();iv!=diapause.end();iv++)
    delete *iv;
  for(vector<Popul*>::iterator iw=adults.begin();iw!=adults.end();iw++)
    delete *iw;

 eggs.clear(); larvae.clear(); pupae.clear(); diapause.clear(); adults.clear();
}

void Populs::Oviposit(int t, double mut){
  for(int i=0;i<5;i++){
    for(vector<Popul*>::iterator it=adults.begin();it!=adults.end();it++){
      if((*it)->Env()==0){
         if(w_beans.size()>0){
           int locate=gsl_rng_uniform_int(randN,w_beans.size());
           if((*it)->Eggs()>0 && (w_beans[locate])->In()<10 && ((w_beans[locate])
->Emerge()==0)){
             eggs.push_back((*it)->Reproduce(t,mut,w_beans[locate]));
             (*it)->Eggs(-1);
             (w_beans[locate])->In(1);
           }
         }
      }
      else{
         int locate=gsl_rng_uniform_int(randN,s_beans.size());
         if((*it)->Eggs()>0 && (s_beans[locate])->In()<10 && (s_beans[locate])->E
merge()==0){
           eggs.push_back((*it)->Reproduce(t,mut,s_beans[locate]));
           (*it)->Eggs(-1);
           (s_beans[locate])->In(1);
         }
      }
    }
  }
}

void Populs::Transform(double T, double *w_T, double *s_T, double *sT, double *d
eltaT){
  int count_w=0;
  int count_s=0;
  *w_T=*s_T=*sT=*deltaT=0.0;
```

```
  for(vector<Bean*>::iterator is=w_beans.begin();is!=w_beans.end();is++)
    (*is)->bTemp(-1.0);

  for(vector<Popul*>::iterator it=larvae.begin();it!=larvae.end();it++){
    if((*it)->Locate()->Env()==0) ((*it)->Locate())->bTemp((*it)->Temp());
    else *sT+=(*it)->Temp();
  }

  for(vector<Popul*>::iterator iu=larvae.begin();iu!=larvae.end();iu++){
    double bT;
    if((*iu)->Age()==9){
      if((*iu)->Locate()->Env()==0) bT=((*iu)->Locate())->bTemp();
      else bT=*sT;
      (*iu)->InitT(T,bT);
    }
    if((*iu)->Age()>9 && (*iu)->Age()<13){
      if((*iu)->Locate()->Env()==0) bT=((*iu)->Locate())->bTemp();
      else bT=*sT;

      (*iu)->Active(T,bT);

      if((*iu)->Age()==12){
         *deltaT+=(*iu)->dT();
         if(((*iu)->Locate())->Env()==0){
           *w_T+=(*iu)->dT();
           count_w+=1;
         }
         if(((*iu)->Locate())->Env()==1){
           *s_T+=(*iu)->dT();
           count_s+=1;
         }
      }
    }
  }
  if(count_w>0) *w_T/=count_w;
  if(count_s>0) *s_T/=count_s;
  if(count_w>0 || count_s>0) *deltaT/=(count_w+count_s);
}

void Populs::Stage(int t, int mode, int *w_E, int *s_E, int *w_L, int *s_L,int *
w_P, int *s_P,int *w_D, int *s_D,int *w_A, int *s_A){
vector<Popul*> baby, kids, young, sleep, old, dead;
*w_E=*s_E=*w_L=*s_L=*w_P=*s_P=*w_D=*s_D=*w_A=*s_A=0;

  if(eggs.size()>0){
    for(vector<Popul*>::iterator is=eggs.begin();is!=eggs.end();is++){
      (*is)->Age(1);
      if((*is)->Locate()->Waste()>0 || gsl_rng_uniform(randN)<0.006){
         dead.push_back(*is);
      }
      else if((*is)->Age()<9){
         baby.push_back(*is);
         if((*is)->Locate()->Env()==0) *w_E+=1;
         else *s_E+=1;
      }
      else{
         kids.push_back(*is);
         if((*is)->Locate()->Env()==0) *w_L+=1;
         else *s_L+=1;
      }
    }
```

```
  }
  if(larvae.size()>0){
    for(vector<Popul*>::iterator it=larvae.begin();it!=larvae.end();it++){
      (*it)->Age(1);
      if((*it)->Locate()->Waste()>0 || gsl_rng_uniform(randN)<0.006){
         dead.push_back(*it);
      }
      else if((*it)->Age()<19){
         kids.push_back(*it);
         if((*it)->Locate()->Env()==0) *w_L+=1;
         else *s_L+=1;
      }
      else{
         young.push_back(*it);
         if((*it)->Locate()->Env()==0) *w_P+=1;
         else *s_P+=1;
      }
    }
  }
  if(pupae.size()>0){
    for(vector<Popul*>::iterator iu=pupae.begin();iu!=pupae.end();iu++){
      (*iu)->Age(1);
      if((*iu)->Locate()->Waste()>0 || gsl_rng_uniform(randN)<0.006){
         dead.push_back(*iu);
      }
      else if((*iu)->Age()<25){
         young.push_back(*iu);
         if((*iu)->Locate()->Env()==0) *w_P+=1;
         else *s_P+=1;
      }
      else{
         (*iu)->Env(((*iu)->Locate())->Env());
         sleep.push_back(*iu);
         ((*iu)->Locate())->Emerge(1);
         if((*iu)->Env()==0) *w_D+=1;
         else *s_D+=1;
      }
    }
  }
  if(diapause.size()>0){
    for(vector<Popul*>::iterator iv=diapause.begin();iv!=diapause.end();iv++){
      (*iv)->Age(1);

      if(gsl_rng_uniform(randN)<0.006) dead.push_back(*iv);
      else if(mode==2 && gsl_rng_uniform(randN)<0.003*((*iv)->Active()) && (*iv)
->Env()==1){
         (*iv)->Env(0);
         (*iv)->Mig(1);
         if((*iv)->Age()<(*iv)->Mature()){
           sleep.push_back(*iv);
           *w_D+=1;
         }
         else{
           old.push_back(*iv);
           *w_A+=1;
         }
      }
      else if((*iv)->Age()<(*iv)->Mature()){
         sleep.push_back(*iv);
         if((*iv)->Env()==0) *w_D+=1;
         else *s_D+=1;
      }
```

```
      else{
         old.push_back(*iv);
         if((*iv)->Env()==0) *w_A+=1;
         else *s_A+=1;
      }
    }
  }
  if(adults.size()>0){
    for(vector<Popul*>::iterator iw=adults.begin();iw!=adults.end();iw++){
      (*iw)->Age(1);
      if(gsl_rng_uniform(randN)<0.006 || (*iw)->Eggs()<=0 || (*iw)->Death()<(*iw
)->Age()) dead.push_back(*iw);
      else{
         old.push_back(*iw);
         if((*iw)->Env()==0) *w_A+=1;
         else *s_A+=1;
      }
    }
  }

  eggs.clear();
  larvae.clear();
  pupae.clear();
  diapause.clear();
  adults.clear();


  for(vector<Popul*>::iterator iw=dead.begin();iw!=dead.end();iw++){
    delete *iw;
  }
  dead.clear();

  while(baby.size()>0){
    eggs.push_back(baby.back());
    baby.pop_back();
  }

  while(kids.size()>0){
    larvae.push_back(kids.back());
    kids.pop_back();
  }

  while(young.size()>0){
    pupae.push_back(young.back());
    young.pop_back();
  }

  while(sleep.size()>0){
    diapause.push_back(sleep.back());
    sleep.pop_back();
  }

  while(old.size()>0){
    adults.push_back(old.back());
    old.pop_back();
  }
  if(adults.size()>0) Shuffle();

  for(vector<Bean*>::iterator ix=waste.begin();ix!=waste.end();ix++){
    delete *ix;
  }
  waste.clear();
```

```
};

void Populs::Shuffle(){
  for (int i=adults.size()-1;i>0;i--){
    int j = gsl_rng_uniform_int(randN,adults.size());
    iter_swap(adults.begin()+i,adults.begin()+j);
  }
};

int Populs::E_size(){
  return eggs.size();
}

int Populs::L_size(){
  return larvae.size();
}

int Populs::P_size(){
  return pupae.size();
}

int Populs::D_size(){
  return diapause.size();
}

int Populs::A_size(){
  return adults.size();
}

int Populs::B_size(){
  return s_beans.size()+w_beans.size();
}

void Populs::Scan(double *w_m_Ww, double *w_m_Th, double *w_v_Ww, double *w_v_Th
, double *w_Pheno, int *w_IL, int *w_PFI, int *w_PLD, int *w_II, int *w_PFD, int
 *w_PLI, int *w_IF, double *s_m_Ww, double *s_m_Th, double *s_v_Ww, double *s_v_
Th, double *s_Pheno, int *s_IL, int *s_PFI, int *s_PLD, int *s_II, int *s_PFD, i
nt *s_PLI, int *s_IF, int *Mig, double *Mig_P){

  *w_m_Ww=*w_m_Th=*w_v_Ww=*w_v_Th=*w_Pheno=0.0;
  *s_m_Ww=*s_m_Th=*s_v_Ww=*s_v_Th=*s_Pheno=0.0;
  *w_IL=*w_PFI=*w_PLD=*w_II=*w_PFD=*w_PLI=*w_IF=0;
  *s_IL=*s_PFI=*s_PLD=*s_II=*s_PFD=*s_PLI=*s_IF=0;
  *Mig=0;

  int w_count=0;
  int s_count=0;

  for(vector<Popul*>::iterator it=adults.begin(); it!=adults.end();it++){
    if((*it)->Env()==0){
      double w=(*it)->Ww();
      double theta=(*it)->Thresh();
      *w_m_Ww+=w; *w_m_Th+=theta; *w_v_Ww+=w*w; *w_v_Th+=theta*theta;
      *w_Pheno+=((*it)->Active());
      *Mig+=((*it)->Mig());
      if((*it)->Mig()==1){
         *Mig_P+=((*it)->Active());
      }

      double x=1.0/(1.0+exp(theta));
      double y=1.0/(1.0+exp(theta-2.0*w));
      double z=1.0/(1.0+exp(theta+2.0*w));
```

```
      if(y-x>0.5) *w_PFI+=1;
      else if(x-z>0.5) *w_PLD+=1;
      else if(z-x>0.5) *w_PFD+=1;
      else if(x-y>0.5) *w_PLI+=1;
      else if(x<0.3) *w_IL+=1;
      else if(x>0.7) *w_IF+=1;
      else *w_II+=1;
      w_count+=1;
    }
    if((*it)->Env()==1){
      double w=(*it)->Ww();
      double theta=(*it)->Thresh();
      *s_m_Ww+=w; *s_m_Th+=theta; *s_v_Ww+=w*w; *s_v_Th+=theta*theta;
      *s_Pheno+=((*it)->Active());

      double x=1.0/(1.0+exp(theta));
      double y=1.0/(1.0+exp(theta-2.0*w));
      double z=1.0/(1.0+exp(theta+2.0*w));
      if(y-x>0.5) *s_PFI+=1;
      else if(x-z>0.5) *s_PLD+=1;
      else if(z-x>0.5) *s_PFD+=1;
      else if(x-y>0.5) *s_PLI+=1;
      else if(x<0.3) *s_IL+=1;
      else if(x>0.7) *s_IF+=1;
      else *s_II+=1;
      s_count+=1;
    }
  }

  for(vector<Popul*>::iterator iu=diapause.begin(); iu!=diapause.end();iu++){
    if((*iu)->Env()==0){
      double w=(*iu)->Ww();
      double theta=(*iu)->Thresh();
      *w_m_Ww+=w; *w_m_Th+=theta; *w_v_Ww+=w*w; *w_v_Th+=theta*theta;
      *w_Pheno+=((*iu)->Active());
      *Mig+=((*iu)->Mig());
      if((*iu)->Mig()==1){
         *Mig_P+=((*iu)->Active());
      }

      double x=1.0/(1.0+exp(theta));
      double y=1.0/(1.0+exp(theta-2.0*w));
      double z=1.0/(1.0+exp(theta+2.0*w));
      if(y-x>0.5) *w_PFI+=1;
      else if(x-z>0.5) *w_PLD+=1;
      else if(z-x>0.5) *w_PFD+=1;
      else if(x-y>0.5) *w_PLI+=1;
      else if(x<0.3) *w_IL+=1;
      else if(x>0.7) *w_IF+=1;
      else *w_II+=1;
      w_count+=1;
    }
    if((*iu)->Env()==1){
      double w=(*iu)->Ww();
      double theta=(*iu)->Thresh();
      *s_m_Ww+=w; *s_m_Th+=theta; *s_v_Ww+=w*w; *s_v_Th+=theta*theta;
      *s_Pheno+=((*iu)->Active());

      double x=1.0/(1.0+exp(theta));
      double y=1.0/(1.0+exp(theta-2.0*w));
      double z=1.0/(1.0+exp(theta+2.0*w));
      if(y-x>0.5) *s_PFI+=1;
```

```
      else if(x-z>0.5) *s_PLD+=1;
      else if(z-x>0.5) *s_PFD+=1;
      else if(x-y>0.5) *s_PLI+=1;
      else if(x<0.3) *s_IL+=1;
      else if(x>0.7) *s_IF+=1;
      else *s_II+=1;
      s_count+=1;
    }
  }

  if(w_count>0){
    *w_m_Ww/=w_count; *w_m_Th/=w_count; *w_v_Ww/=w_count; *w_v_Th/=w_count;
    *w_Pheno/=w_count;
    *w_v_Ww-=(*w_m_Ww)*(*w_m_Ww); *w_v_Th-=(*w_m_Th)*(*w_m_Th);
  }
  if(s_count>0){
    *s_m_Ww/=s_count; *s_m_Th/=s_count; *s_v_Ww/=s_count; *s_v_Th/=s_count;
    *s_Pheno/=s_count;
    *s_v_Ww-=(*s_m_Ww)*(*s_m_Ww); *s_v_Th-=(*s_m_Th)*(*s_m_Th);
  }

  if(*Mig>0){
  *Mig_P/=*Mig;
  }
}

void Populs::Show(){
  for(vector<Popul*>::iterator is=diapause.begin();is!=diapause.end();is++){
    cout<<(*is)->Ww()<<" "<<(*is)->Thresh()<<" "<<(*is)->Active()<<" "<<(*is)->En
v()<<" "<<(*is)->Mig()<<" "<<(*is)->Vol()<<endl;
  }
  for(vector<Popul*>::iterator it=adults.begin();it!=adults.end();it++){
    cout<<(*it)->Ww()<<" "<<(*it)->Thresh()<<" "<<(*it)->Active()<<" "<<(*it)->En
v()<<" "<<(*it)->Mig()<<" "<<(*it)->Vol()<<endl;
  }
}

void Populs::Show(int t){
  int w_bE, w_bL, w_bP, w_bD, w_bA, s_bE, s_bL, s_bP, s_bD, s_bA;
  int w_tE, w_tL, w_tP, w_tD, w_tA, s_tE, s_tL, s_tP, s_tD, s_tA;
  int w_mE, w_mL, w_mP, w_mD, w_mA, s_mE, s_mL, s_mP, s_mD, s_mA;
  int mig_w_bD, mig_w_bA, mig_w_tD, mig_w_tA, mig_w_mD, mig_w_mA;

  w_bE=w_bL=w_bP=w_bD=w_bA=s_bE=s_bL=s_bP=s_bD=s_bA=0;
  w_tE=w_tL=w_tP=w_tD=w_tA=s_tE=s_tL=s_tP=s_tD=s_tA=0;
  w_mE=w_mL=w_mP=w_mD=w_mA=s_mE=s_mL=s_mP=s_mD=s_mA=0;
  mig_w_bD=mig_w_bA=mig_w_tD=mig_w_tA=mig_w_mD=mig_w_mA=0;

  for(vector<Popul*>::iterator is=eggs.begin();is!=eggs.end();is++){
    if(((*is)->Locate())->Env()==0){
      if(1.0 < (*is)->Ww()){
         if(0.0 < (*is)->Thresh() && (*is)->Thresh() < 1.0) w_bE+=1;
         else if((*is)->Thresh() < 2.0) w_tE+=1;
         else w_mE+=1;
      }
    }
    else{
      if(1.0 < (*is)->Ww()){
         if (0.0 < (*is)->Thresh() && (*is)->Thresh() < 1.0) s_bE+=1;
         else if ((*is)->Thresh() < 2.0) s_tE+=1;
         else s_mE+=1;
      }
```

```
    }
  }

  for(vector<Popul*>::iterator it=larvae.begin();it!=larvae.end();it++){
    if(((*it)->Locate())->Env()==0){
      if(1.0 < (*it)->Ww()){
         if(0.0 < (*it)->Thresh() && (*it)->Thresh() < 1.0) w_bL+=1;
         else if((*it)->Thresh() < 2.0) w_tL+=1;
         else w_mL+=1;
      }
    }
    else{
      if(1.0 < (*it)->Ww()){
         if(0.0 < (*it)->Thresh() && (*it)->Thresh() < 1.0)    s_bL+=1;
         else if((*it)->Thresh() < 2.0)    s_tL+=1;
         else    s_mL+=1;
      }
    }
  }

  for(vector<Popul*>::iterator iw=pupae.begin();iw!=pupae.end();iw++){
    if(((*iw)->Locate())->Env()==0){
      if(1.0 < (*iw)->Ww()){
         if(0.0 < (*iw)->Thresh() && (*iw)->Thresh() < 1.0)  w_bP+=1;
         else if((*iw)->Thresh() < 2.0)  w_tP+=1;
         else   w_mP+=1;
      }
    }
    else{
      if(1.0 < (*iw)->Ww()){
         if(0.0 < (*iw)->Thresh() && (*iw)->Thresh() < 1.0)     s_bP+=1;
         else if((*iw)->Thresh() < 2.0)     s_tP+=1;
         else    s_mP+=1;
      }
    }
  }

  for(vector<Popul*>::iterator iu=diapause.begin();iu!=diapause.end();iu++){
    if((*iu)->Env()==0){
      if(1.0 < (*iu)->Ww()){
         if((*iu)->Mig()==0){
           if(0.0 < (*iu)->Thresh() && (*iu)->Thresh() < 1.0)  w_bD+=1;
           else if((*iu)->Thresh() < 2.0)  w_tD+=1;
           else   w_mD+=1;
         }
         else{
           if(0.0 < (*iu)->Thresh() && (*iu)->Thresh() < 1.0)  mig_w_bD+=1;
           else if((*iu)->Thresh() < 2.0)  mig_w_tD+=1;
           else   mig_w_mD+=1;
         }
      }
    }
    else{
      if(1.0 < (*iu)->Ww()){
         if(0.0 < (*iu)->Thresh() && (*iu)->Thresh() < 1.0)     s_bD+=1;
         else if((*iu)->Thresh() < 2.0)     s_tD+=1;
         else  s_mD+=1;
      }
    }
  }

  for(vector<Popul*>::iterator iv=adults.begin();iv!=adults.end();iv++){
```

```
    if((*iv)->Env()==0){
      if(1.0 < (*iv)->Ww()){
        if((*iv)->Mig()==0){
          if (0.0 < (*iv)->Thresh() && (*iv)->Thresh() < 1.0)     w_bA+=1;
          else if ((*iv)->Thresh() < 2.0)     w_tA+=1;
          else   w_mA+=1;
        }
        else{
          if (0.0 < (*iv)->Thresh() && (*iv)->Thresh() < 1.0)     mig_w_bA+=1;
          else if ((*iv)->Thresh() < 2.0)     mig_w_tA+=1;
          else  mig_w_mA+=1;
        }
      }
    }
    else{
      if(1.0 < (*iv)->Ww()){
        if(0.0 < (*iv)->Thresh() && (*iv)->Thresh() <1.0)     s_bA+=1;
        else if((*iv)->Thresh() <2.0)     s_tA+=1;
        else   s_mA+=1;
      }
    }
  }

  cout<<t<<" "<<w_bE<<" "<<w_bL+w_bP<<" "<<w_bD<<" "<<w_bA<<" "<<w_tE<<" "<<w_tL+w_
tP<<" "<<w_tD<<" "<<w_tA<<" "<<w_mE<<" "<<w_mL+w_mP<<" "<<w_mD<<" "<<w_mA<<" "<<s_bE
<<" "<<s_bL+s_bP<<" "<<s_bD<<" "<<s_bA<<" "<<s_tE<<" "<<s_tL<<" "<<s_tD+s_tP<<" "<<s
_tA<<" "<<s_mE<<" "<<s_mL<<" "<<s_mD+s_mP<<" "<<s_mA<<" "<<mig_w_bD<<" "<<mig_w_bA<
<" "<<mig_w_tD<<" "<<mig_w_tA<<" "<<mig_w_mD<<" "<<mig_w_mA<<endl;

}

void Populs::Harvest(int B, int t){
  if(t%360<180){
    for(int i=0; i<B; i++){
      Bean *b = new Bean(0);
      w_beans.push_back(b);
    }
  }
}

void Populs::Turn_over(int mode, int t){
  vector<Bean*> wild;

  if(mode > 0){
    B_Shuffle();
    vector<Bean*> store(s_beans.begin()+10,s_beans.end());
    for(int j=0; j<10; j++){
      (s_beans[j])->Waste(1);
      waste.push_back(s_beans[j]);
    }
    s_beans.clear();

    if(mode < 3 && t%360<180){
      for(int h=0; h<5; h++){
        Bean *b = new Bean(1);
        store.push_back(b);
        (w_beans[w_beans.size()-1])->Env(1);
        store.push_back(w_beans.back());
        w_beans.pop_back();
      }
    }
    else{
```

```
      for(int h=0; h<10; h++){
        Bean *b = new Bean(1);
        store.push_back(b);
      }
    }

    while(store.size()>0){
      s_beans.push_back(store.back());
      store.pop_back();
    }
    store.clear();
  }

  if(mode < 3 && w_beans.size()>0){
    if(t%360==240){
      for(vector<Bean*>::iterator ir=w_beans.begin();ir!=w_beans.end();ir++){
        (*ir)->Waste(1);
        waste.push_back(*ir);
      }
      w_beans.clear();
    }
  }
}

void Populs::B_Shuffle(){
  for (int i=w_beans.size()-1;i>0;i--){
    int j = gsl_rng_uniform_int(randN,w_beans.size());
    iter_swap(w_beans.begin()+i,w_beans.begin()+j);
  }
  for (int h=s_beans.size()-1;h>0;h--){
    int k = gsl_rng_uniform_int(randN,s_beans.size());
    iter_swap(s_beans.begin()+h,s_beans.begin()+k);
  }
}
```

05年 12月12日 21:20 popul.h 1/1 ページ

```
#ifndef POPUL_H
#define POPUL_H

#include <vector>
#include <iostream>
#include "bean.h"
#include <math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>

using namespace std;
extern gsl_rng *randN;

class Bean;

class Popul{
 private:
  int env,age,mature,eggs,death,mig,vol;
  double P,w_w,S;
  double initT,aveT,temp;
  Bean *bean;

 public:
  Popul(int);
  Popul(int,double,double,Bean*);
  ~Popul(){};
  Popul* Reproduce(int,double,Bean*);

  double Ww();
  double Thresh();
  double Temp();
  double dT();
  int Age();
  int Mature();
  int Vol();
  double Active();
  int Eggs();
  int Death();
  int Env();
  int Mig();

  void Env(int);
  void InitT(double, double);
  void Eggs(int);
  void Age(int);
  void Active(double,double);
  void Mig(int);
  Bean* Locate();
};

#endif //POPUL_H
```

```
#include "popul.h"

Popul::Popul(int mode){
  w_w=10.0*gsl_rng_uniform(randN)-5.0;
  S=20.0*gsl_rng_uniform(randN)-10.0;

  P=1.0; initT=0.0; aveT=0.0;
  age = 360; mature = 340; death = 370; eggs = 10;
  mig=0; vol=-1;
  if(mode<3) env=0;
  else env=1;
};

Popul::Popul (int v,double w_1, double s_1, Bean *b){
  w_w = w_1; S = s_1;
  P=1.0; env=0; bean=b;
  initT=0.0; aveT=0.0; age = 0; mature=340; death = 370; eggs = 10; mig=0;
  if(v>=0) vol=v+1;
  else vol=-1;
};

Popul* Popul::Reproduce(int t,double u, Bean *B){
  double w_1, s_1;
  int v;
  Popul *kid;

  w_1 = w_w;  s_1 = S;
  v=vol;

  double mut;
  if(gsl_rng_uniform(randN) < u){
    mut = gsl_ran_gaussian(randN,0.1);
    w_1+=mut;
  }
  if(gsl_rng_uniform(randN) < u){
    mut = gsl_ran_gaussian(randN,0.2);
    s_1+=mut;
  }

  if (w_1>5.0) w_1=5.0;
  else if (w_1<-5.0) w_1=-5.0;
  if (s_1>10.0) s_1=10.0;
  else if (s_1<-10.0) s_1=-10.0;

  if(v<0 && t>=17640) v=0;

  kid = new Popul(v, w_1, s_1, B);
  return kid;
};

double Popul::Ww(){
  return w_w;
};

double Popul::Thresh(){
  return S;
};

void Popul::Active(double eT, double T){
  aveT+=T+eT;

  if(age == 12){
```

```
    aveT/=3.0;
    double dT=aveT-initT;
    double out=0.0;

    out=1.0/(1.0+exp(S-w_w*dT));
    P=out;
    mature=25+int(315*out);
    death=30+mature;
    eggs=int(90*(1.0-out)+10);
  }
};

double Popul::Active(){
  return P;
}

Bean* Popul::Locate(){
  return bean;
}

int Popul::Age(){
  return age;
}

int Popul::Mature(){
  return mature;
}

int Popul::Vol(){
  return vol;
}

double Popul::dT(){
  return aveT-initT;
}

int Popul::Mig(){
  return mig;
}

void Popul::Mig(int m){
  mig=m;
}

void Popul::Age(int t){
  age+=t;
  if(age > 371) cout<<age <<endl;
}

double Popul::Temp(){
  if(age>11 && age<19)  return 0.0055-0.00055*(double(age-15))*(double(age-15));
  else return 0.0;
}

int Popul::Eggs(){
  return eggs;
}

void Popul::Eggs(int e){
  eggs+=e;
}
```

```
int Popul::Env(){
  return env;
}

void Popul::Env(int e){
  env=e;
}

int Popul::Death(){
  return death;
}

void Popul::InitT(double T, double eT){
  initT = T+ eT;
}
```

```
#ifndef BEAN_H
#define BEAN_H

#include <vector>
#include <algorithm>

using namespace std;

class Bean{
 private:
  double b_temp;
  int in, emerge, waste, env;

  public:
  Bean(int);
  ~Bean();
  void In(int);
  int In();
  int Env();
  void Env(int);
  int Waste();
  void Waste(int);
  int Emerge();
  void Emerge(int );
  void bTemp(double);
  double bTemp();
};

#endif //BEAN_H
```

```
#include "bean.h"

Bean::Bean(int e){
  b_temp=0.0; in=0; emerge=0; env=e; waste=0;
}

Bean::~Bean(){
}

void Bean::In(int n){
  in+=n;
}

int Bean::In(){
  return in;
}

void Bean::Env(int n){
  env=n;
}

int Bean::Env(){
  return env;
}

void Bean::Waste(int n){
  waste=n;
}

int Bean::Waste(){
  return waste;
}

void Bean::bTemp(double t){
  if(t<0) b_temp=0.0;
  else b_temp+=t;
}

double Bean::bTemp(){
  return b_temp;
}

int Bean::Emerge(){
  return emerge;
}

void Bean::Emerge(int e){
  emerge+=e;
}
```

References

Alvarez, N., D. Mckey, M. Hossaert-Mckey, C. Born, L. Mercier, and B. Benrey (2005). Ancient and recent evolutionary history of the bruchid beetle, *Acanthoscelides obtectus* Say, a cosmopolitan pest of beans. *Molecular Ecology 14*, 1015–1024.

Applebaum, S. W. and Y. Heifetz (1999). Density-dependent physiological phase in insects. *Annual Review of Entomology 44*, 317–341.

Appleby, J. H. and P. F. Credland (2001). Bionomics and polymorphism in *Callosobruhus subinnotatus. Bulletin of Entomological Research 91*, 235–244.

BBC Weather (2003). Average conditions of Niamy, Niger. `http://www.bbc.co.uk/weather/world/city_guides/results.shtml?tt=TT000480`.

Bell, C. H. (1994). A review of diapause in stored-product insects. *Journal of Stored Products Research 30*, 99–120.

Blarer, A. and M. Doebeli (1999). Resonance effects and outbreaks in ecological time series. *Ecology Letters 2*, 167–177.

Caswell, G. H. (1960). Observations on an abnormal form of *Callosobruchus maculatus* (F.). *Bulletin of Entomological Research 50*, 671–680.

Dingle, H. and R. Winchell (1997). Juvenile hormone as a mediator of plasticity in insect life histories. *Archives of Insect Biochemistry and Physiology 35*, 359–373.

Fairbairn, D. J. (1994). Wing dimorphism and the migratory syndrome: correlated traits for migratory tendency in wing dimorphic insects. *Researches on Population Ecology 36*, 157–163.

Fielding, D. J. (2004). Intraspecific competition and spatial heterogeneity alter life history traits in an individual-based model of grasshoppers*. *Ecological Modelling 175*, 169–187.

Galassi, M., D. J. Theiler, B. Gough, J. G., M. Booth, and F. Rossi (2003). *GNU Scientific Library Reference Manual - Second Edition*. Network Theory Limited.

George, J. and K. K. Verma (1994). Polymorphism in *Callosobruchus maculatus* (F.) (Coleoptera, Bruchidae) - new dimensions. *Russian Entomol. J. 3*, 93–107.

Glitho, I. A., A. Lenga, D. Pierre, and J. Huignard (1996). Changes in the responsiveness during two phases of diapause termination in *Bruchidius atrolineatus* Pic (Coleoptera: Bruchidae). *Journal of Insect Physiology 10*, 953–960.

Harrison, R. G. (1980). Dispersal polymorphisms in insects. *Annual Review of Ecology and Systematics 11*, 95–118.

Hodek, I., A. Bonet, and Hodková (1981). Some ecological factors affecting diapause in adults of *Acanthoscelides obtectus* from Mexican mountains. In L. V. (Ed.), *The Ecology of Bruchids Attacking Legumes (pulses).* Dr W. Junk Publishers, The Hague.

Hunt, J. H. and G. V. Amdam (2005). Bivoltinism as an antecedent to eqsociality in the paper wasp genus *Polistes*. *Science 308*, 264–267.

Kiritani, K. (1961). The origin of the household pests, and the process of adaptation to indoor conditions. (in Japanese). *Seitai Konchu 9*, 22–40.

Lenga, A., C. Thibeaudeau, and J. Huignard (1991). Influence of thermoperiod and photoperiod on reproductive diapause in *Bruchidius atrolineatus* (Pic) (Coleoptera, Bruchidae). *Physiological Entomology 16*, 295–303.

Lextrait, P., J. C. Biemont, and J. Pouzat (1995). Pheromone release by two forms of *Callosobruchus maculatus* females: effect of age, temperature and host plant. *Physiological Entomology 20*, 309–317.

Monge, J. P. and J. Huignard (1991). Population fluctuations of two bruchid species *Callosobruchus maculatus* (F.) and *Bruchidius atrolineatus* (Pic) (Coleoptera Bruchidae) and their parasitoids *Dinarmus basalis* (Rondani) and *Eupelmus vuilleti* (Crawford) (Hymenoptera, Pteromalidae, Eupelmidae) in a storage situation in Niger. *Journal of African Zoology 105*, 187–196.

Morse, G. E. and B. D. Farrell (2005). Ecological and evolutionary diversification of the seed beetle genus *Stator* (Coleoptera: Chrysomelidae: Bruchinae). *Evolution 59*, 1315–1333.

Pichard, B., B. Leroi, and A. Bonet (1991). Comparison des cycles d'*Acanthoscelides obtectus* et d'*A. obvelatus* (coleoptères, bruchidae) à tepoztlan (mexique). *Acta Oecologica 12*, 185–201.

Pigliucci, M. and C. Murren (2003). Genetic assimilation and a possible evolutionary paradox: can macroevolution sometimes be so fast as to pass us by? *Evolution 57*, 1455–1464.

Roff, D. A. (1986). The evolution of wing dimorphism in insects. *Evolution 40*, 1009–1020.

Roff, D. A. (1994). Habitat persistence and the evolution of wing dimorphism in insects. *American Naturalist 144*, 772–798.

Sano, I. (1967). Density effect and environmental temperature as the factors producing the active form of *Callosobruchus maculatus* (F.) (Coleoptera, Bruchidae). *Journal of Stored Products Research 2*, 187–195.

Sano-Fujii, I. (1979). Effect of parental age and developmental rate on the production of active form of *Callosobruchus maculatus* (F.) (Coleoptera: Bruchidae). *Mechanisms of Ageing and Development 10*, 283–293.

Sano-Fujii, I. (1984). Effect of bean water content on the production of the active form of *Callosobruchus maculatus* (F.) (Coleoptera:

Bruchidae). *Journal of Stored Products Research 20*, 153–161.

Sano-Fujii, I. (1986). The genetic basis of the production of the active form of *Callosobruchus maculatus* (F.) (Coleoptera: Bruchidae). *Journal of Stored Products Research 22*, 115–123.

Schmalhausen, I. I. (1949). *Factors of Evolution: The Theory of Stabilizing Selection.* Blakiston, Philadelphia, PA.

Strand, E., G. Huse, and J. Giske (2002). Artificial evolution of life history and behavior. *American Naturalist 159*, 624–244.

Takano, M., Y. Toquenaga, and K. Fujii (2001). Polymorphism of competition type and its genetics in *Callosobruchus maculatus* (Coleoptera: Bruchidae). *Population Ecology 43*, 265–273.

Tatar, M. and C.-M. Yin (2001). Slow aging during insect reproductive diapause: whiy butterflies, grasshoppers and flies are like worms. *Experimental Gerontology 36*, 723–738.

Tran, B., J. Darquenne, and J. Huignard (1993). Changes in responsiveness to factors inducing diapause termination in *Bruchus rufimanus* (Boh.) (Coleoptera: Bruchidae). *Journal of Insect Physiology 39*, 769–774.

Tran, B. and J. Huignard (1992). Interactions between photoperiod and food affect the termination of reproductive diapause in *Bruchus rufimanus* (Boh.), (Coleoptera, Bruchidae). *Journal of Insect Physiology 38*, 633–642.

Utida, S. (1954). Phase dimorphism observed in the laboratory population of cowpea weevil, *Callosobruchus quadrimaculatus. Japanese Journal of Applied Zool. 18*, 161–168. (In Japanese with English summary).

Utida, S. (1956). Phase dimorphism observed in the laboratory population of cowpea weevil, *Callosobruchus quadrimaculatus.* II. Differential effects of temperature, humidity and population density upon some ecological characters of the two phases. *Researches on Population Ecology, Kyoto 3*, 93–104 (In Japanese with English summary).

Utida, S. (1972). Density-dependent polymorphism in the adult of *Callosobruchus maculatus. Journal of Stored Products Research 8*, 111–126.

Utida, S. (1981). Polymorphism and phase dimorphism in *Callosobruchus.* In L. V. (Ed.), *The Ecology of Bruchids Attacking Legumes (pulses).* Dr W. Junk Publisher, The Hague.

Utida, S. (1998). *Ecology of Animal Populations. (in Japanese).* Kyoto University Press.

Watanabe, N. (1990). Diversity in life cycle patterns of bruchids occurring in Japan (Coleoptera: Bruchidae). In K. Fujii, A. M. R. Gatehouse, C. D. Johnson, R. Mitchel, and T. Yoshida (Eds.), *Bruchids and Legumes: Economics, Ecology and Coevolution.* Kluwer

Academic Publishers.

Wiener, P. and S. Tuljapurkar (1994). Migration in variable environments: exploring life-history evolution using structured population moadels. *Journal of Theoretical Biology 166*, 75–90.

Zannou, E. T., I. A. Glitho, J. Huignard, and J. P. Monge (2003). Life history of flight morph females of *Callosobruchus maculatus* F.: evidence of a reproductive diapause. *Journal of Insect Physiology 49*, 575–582.

Table 1.1 Comparison of life history traits and adaptation to the storage in Bruchidae. (After Kiritani 1961, Utida 1998.)

	Type		
	Field type	Intermediate type	Storage type
Reproductive diapause	Long	Plastic(Long/Short)	Short
Flight ability	High	Plastic(High/Low)	Low
Fecundity	Low	Plastic(High/Low)	Low
Digestive ability	Raw beans	Raw and dry beans	Dry beans

Table 1.2 Correspondence table of classification by Kiritani (1961) and Watanabe (1990).

	Digestive ability		
	Raw beans	Raw and dry beans	Dry beans
my paper	Field type	Intermediate type	Storage type
Kiritani (1961)	Field type	Cross type	Storage type
Watanabe (1990)	Field bruchids	SH-type strain in storage bruchids	H-type strain in storage bruchids

Table 3.1 The parameter settings of life history traits in my model.

Life history trait	Parameter
Sensitive age	$9-12$-th days
Infuential age	$12-18$-th days
Age at emergence	25-th day
Age at maturity	$25+315P(dT)$-th day
Maximum life time	$55+315P(dT)$-th day
Fecundity	$10+90(1.0-P(dT))$
Migration rate from the storage to the field	$0.003P(dT)$ per day

Table 3.2 The seven types based on the norm of reaction.

Type	Condition
PFI	$P(2) - P(0) > 0.5$
PLD	$P(0) - P(-2) > 0.5$
PLI	$P(0) - P(2) > 0.5$
PFD	$P(-2) - P(0) > 0.5$
IL	$\|P(2) - P(0)\| < 0.5$ and $\|P(0) - P(-2)\| < 0.5$ and $P(0) < 0.3$
II	$\|P(2) - P(0)\| < 0.5$ and $\|P(0) - P(-2)\| < 0.5$ and $0.3 \leq P(0) \leq 0.7$
IF	$\|P(2) - P(0)\| < 0.5$ and $\|P(0) - P(-2)\| < 0.5$ and $0.7 < P(0)$

Table 4.1 The number of cases where each type dominated in simulation of asexual reproduction model in the field condition. The number within parenthesis represents the number of cases where each type was fixed by the 18000-th day.

PFI	PLD	PLI	PFD	IL	II	IF	GP
86	0	0	6	0	0	4	4
(15)	(0)	(0)	(0)	(0)	(0)	(0)	

Table 4.2 The number of cases where each type dominated in simulation of sexual reproduction model in the field condition. The number within parenthesis represents the number of cases where each type was fixed by the 18000-th day.

PFI	PLD	PLI	PFD	IL	II	IF	GP
69	1	0	0	0	0	8	22
(16)	(0)	(0)	(0)	(0)	(0)	(0)	

Table 5.1 The number of cases where each type dominated in simulation of asexual reproduction model in heterogeneous condition of field and storage. The number within parenthesis represents the number of cases where each type was fixed in each condition by the 18000-th day. Ext means local extinction by the 18000-th day.

	PFI	PLD	PLI	PFD	IL	II	IF	GP	Ext
Field	79	0	0	13	2[a]	2	0	3	1
	(19)	(0)	(0)	(11)	(2)	(0)	(0)		
Storage	29	0	0	7	35	0	0	29	0
	(7)	(0)	(0)	(3)	(0)	(0)	(0)		

[a] The fixation of IL in the field was the special case where local extinction occured in the field, and only one or two individials migrated from the storage to the field, though IL rarely migrated to the field.

Table 5.2 The number of cases where each type dominated in simulation of asexual reproduction model without migration in heterogeneous condition of field and storage. The number within parenthesis represents the number of cases where each type was fixed in each condition by the 18000-th day. Ext means local extinction by the 18000-th day.

	PFI	PLD	PLI	PFD	IL	II	IF	GP	Ext
Field	70	0	0	6	0	2	0	1	21
	(12)	(0)	(0)	(0)	(0)	(1)	(0)		
Storage	0	0	0	0	42	0	0	58	0
	(0)	(0)	(0)	(0)	(11)	(0)	(0)		

Table 5.3 The number of cases where each type dominated in simulation of sexual reproduction model in heterogeneous condition of field and storage. The number within parenthesis represents the number of cases where each type was fixed in each condition by the 18000-th day. Ext means local extinction by the 18000-th day.

	PFI	PLD	PLI	PFD	IL	II	IF	GP	Ext
Field	92 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	8	0
Storage	94 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	6	0

Table 5.4 The number of cases where each type dominated in simulation of sexual reproduction model without migration in heterogeneous condition of field and storage. The number within parenthesis represents the number of cases where each type was fixed in each condition by the 18000-th day. Ext means local extinction by the 18000-th day.

	PFI	PLD	PLI	PFD	IL	II	IF	GP	Ext
Field	81	0	0	0	0	0	0	7	12
	(1)	(0)	(0)	(0)	(0)	(0)	(0)		
Storage	78	0	0	0	11	0	0	11	0
	(0)	(0)	(0)	(0)	(1)	(0)	(0)		

Table 6.1 The number of cases where each type dominated in simulation of asexual reproduction model in the storage condition. The number within parenthesis represents the number of cases where each type was fixed by the 18000-th day.

PFI	PLD	PLI	PFD	IL	II	IF	GP
0	0	0	0	100	0	0	0
(0)	(0)	(0)	(0)	(65)	(0)	(0)	

Table 6.2 The number of cases where each type dominated in simulation of sexual reproduction model in the storage condition. The number within parenthesis represents the number of cases where each type was fixed by the 18000-th day.

PFI	PLD	PLI	PFD	IL	II	IF	GP
0	0	0	0	95	0	0	5
(0)	(0)	(0)	(0)	(0)	(0)	(0)	

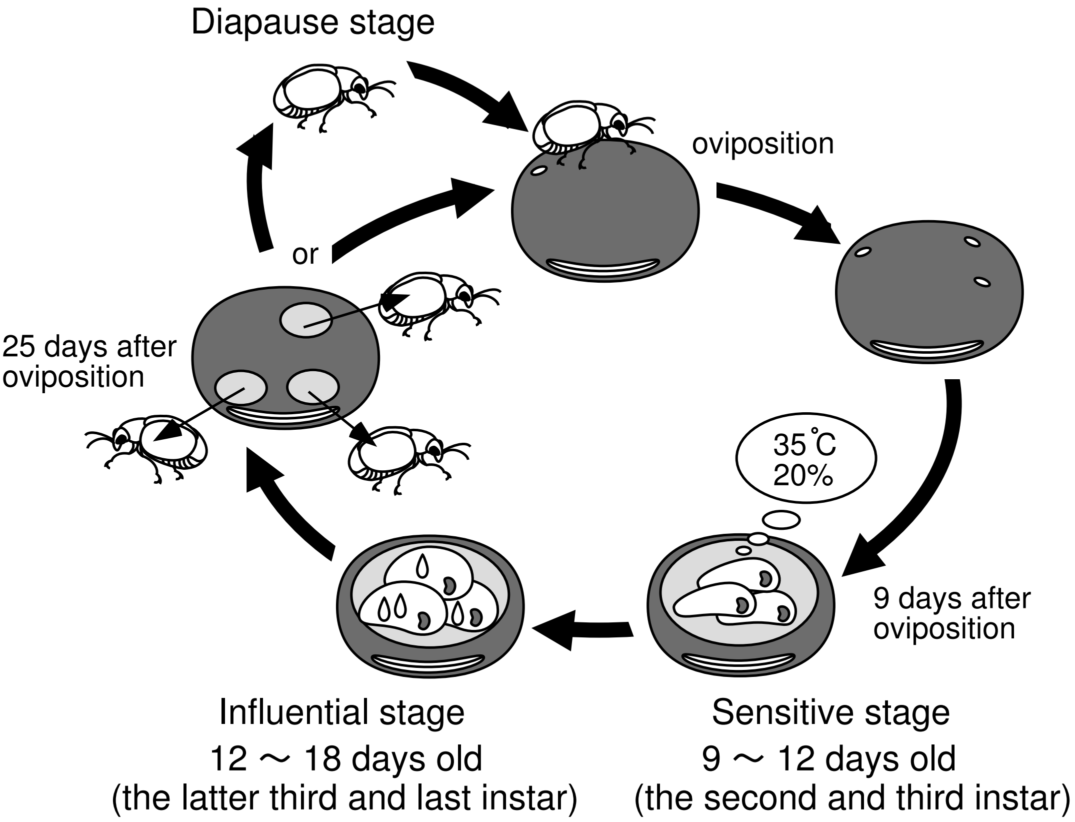


FIGURE 2.1. The life cycle of *C. maculatus*.

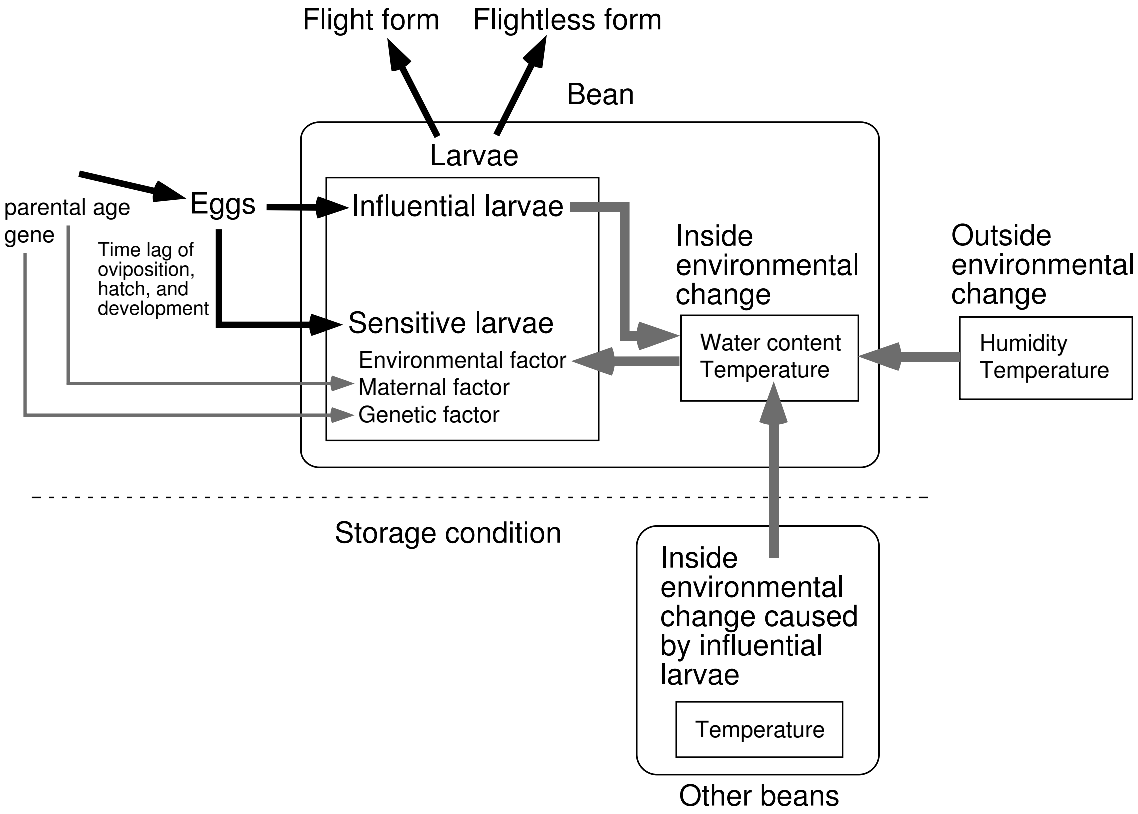


Figure 2.2. The main factors to affect the probability of emergence of flight form.

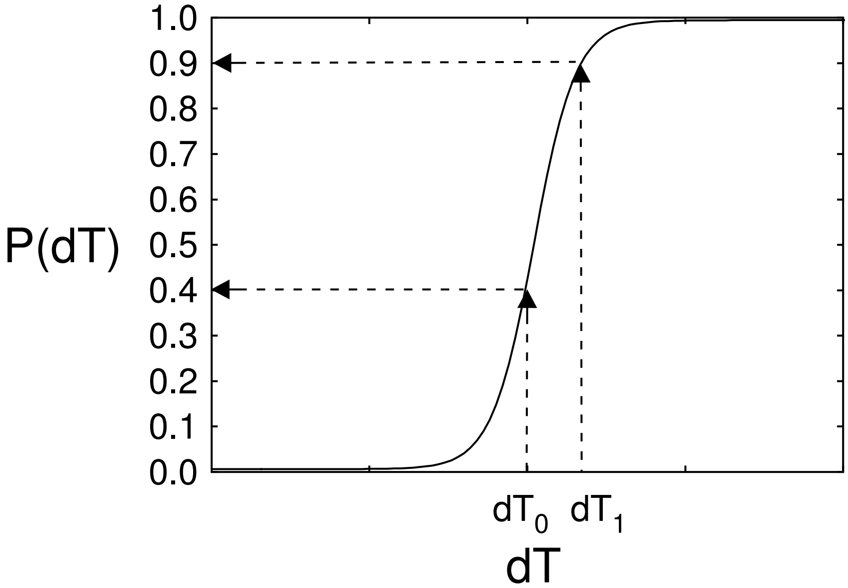


FIGURE 3.1. One example of the sigmoid function used for reaction norms. This sigmoid function represents that the liability to close to flight form is 0.4 under no change of temperature (dT_0), and the liability increases to 0.9 when the change of temperature is dT_1.

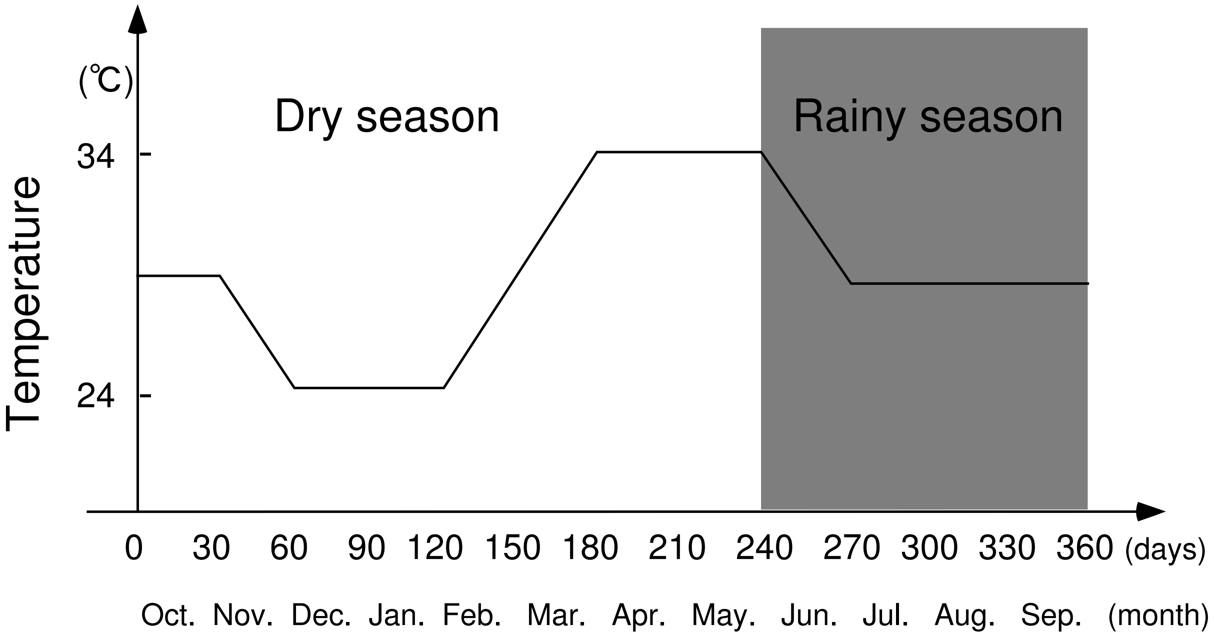


FIGURE 3.2 The model curve of seasonal change of outside temperature. The curve was based on Niamey region in Niger.

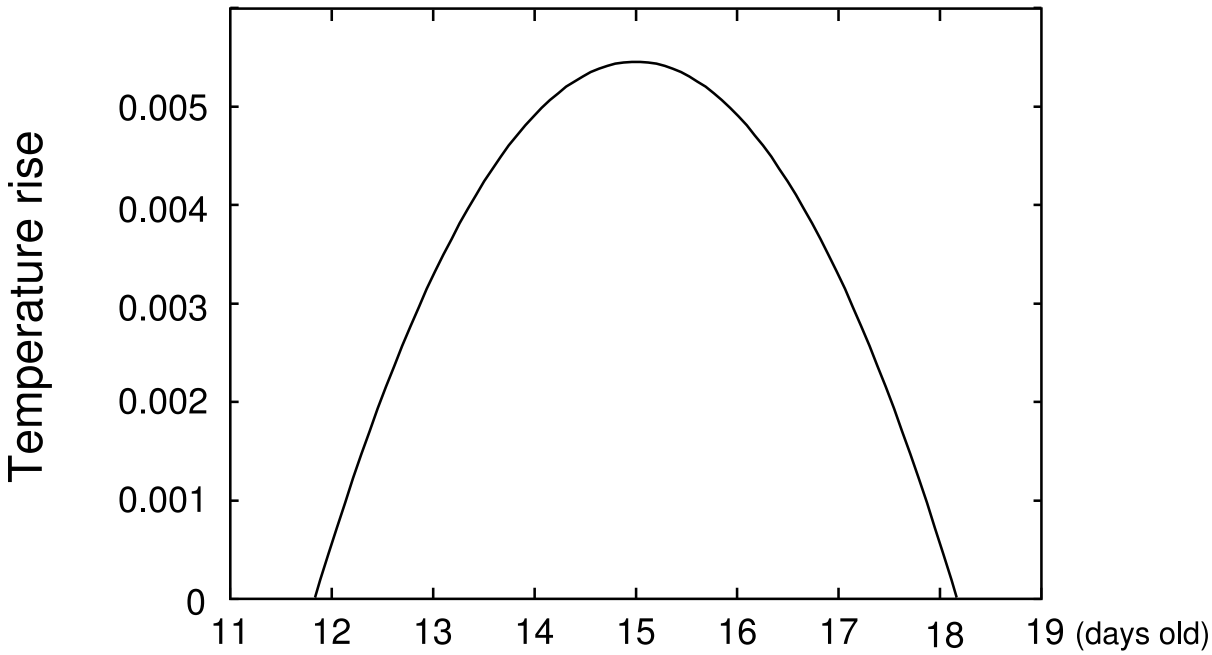


FIGURE 3.3. The model curve of inside temperature caused by an influential larva.

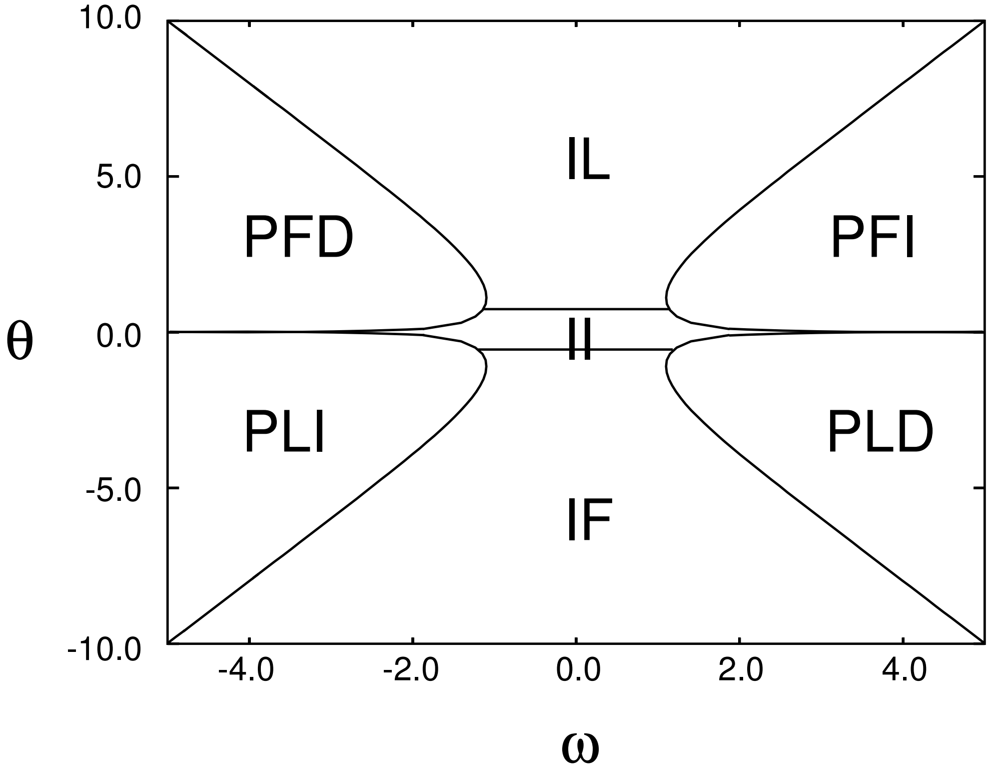


FIGURE 3.4. The parameter region of each type.

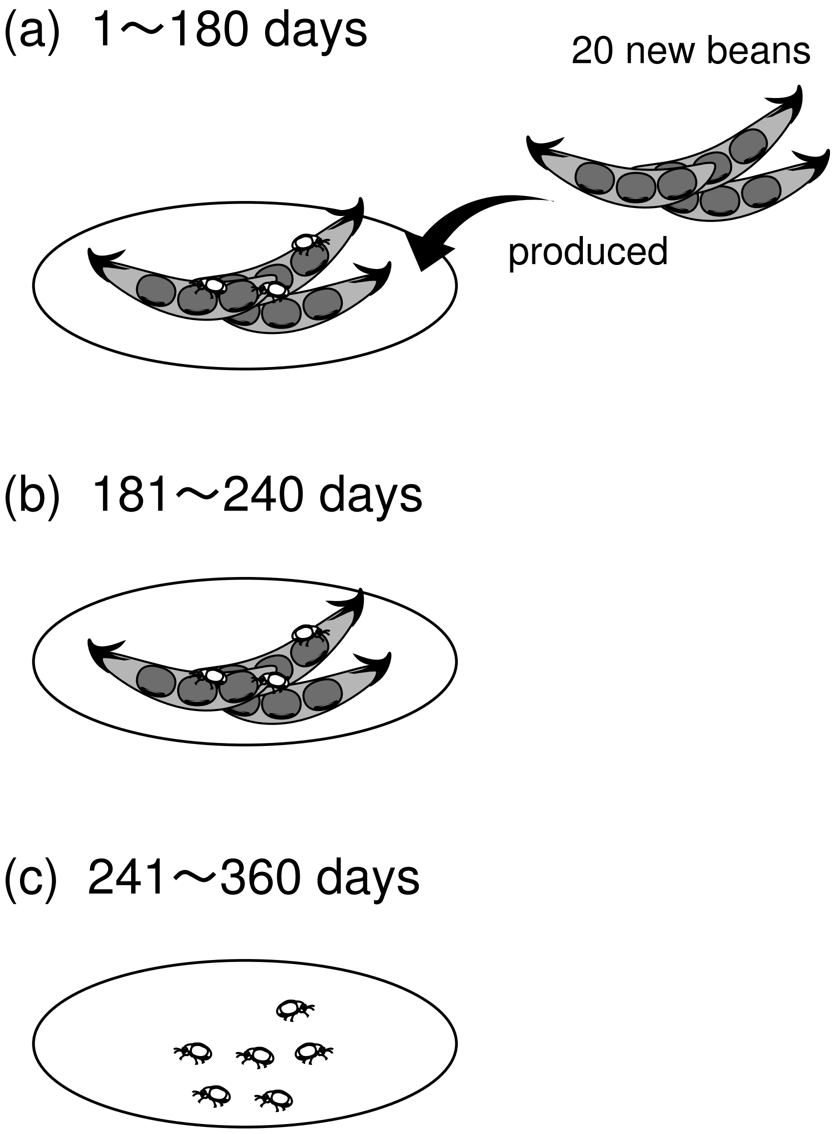


FIGURE 4.1 Resource dynamics in the field condition. (a) harvest-time; (b) before rainy season; (c) rainy season.

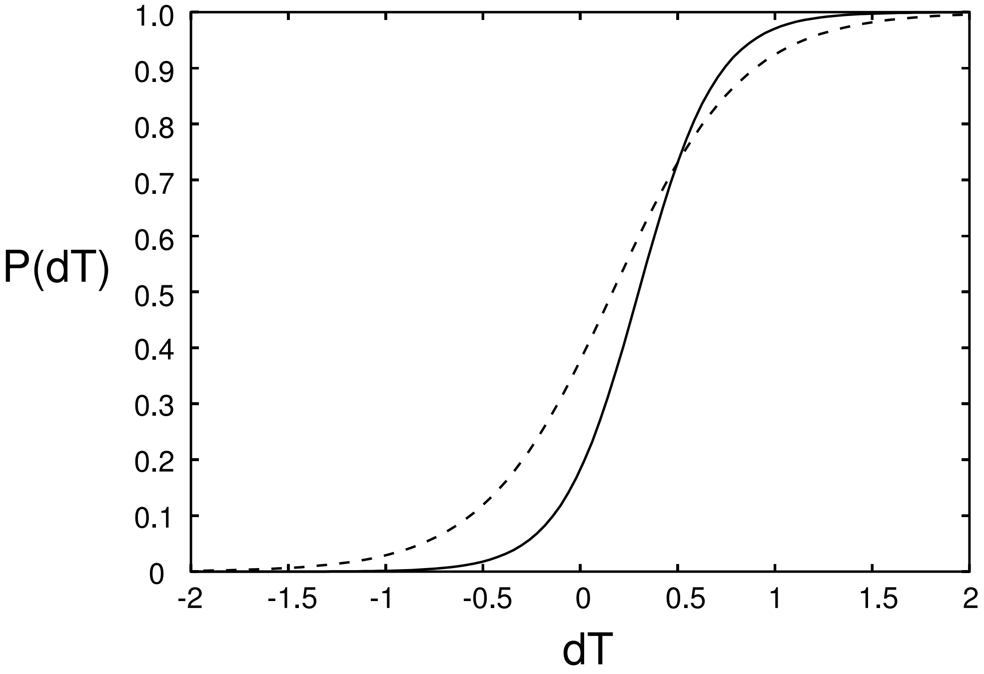


FIGURE 4.2 Examples of typical reaction norms in simulation of asexual reproduction model in the field condition. The reaction norms of bPFI (broken line) and tPFI (solid line) showed bi-voltinism and tri-voltinism, respectively. bPFI $(\omega, \theta) = (3.0, 0.5)$; tPFI $(\omega, \theta) = (5.0, 1.5)$.

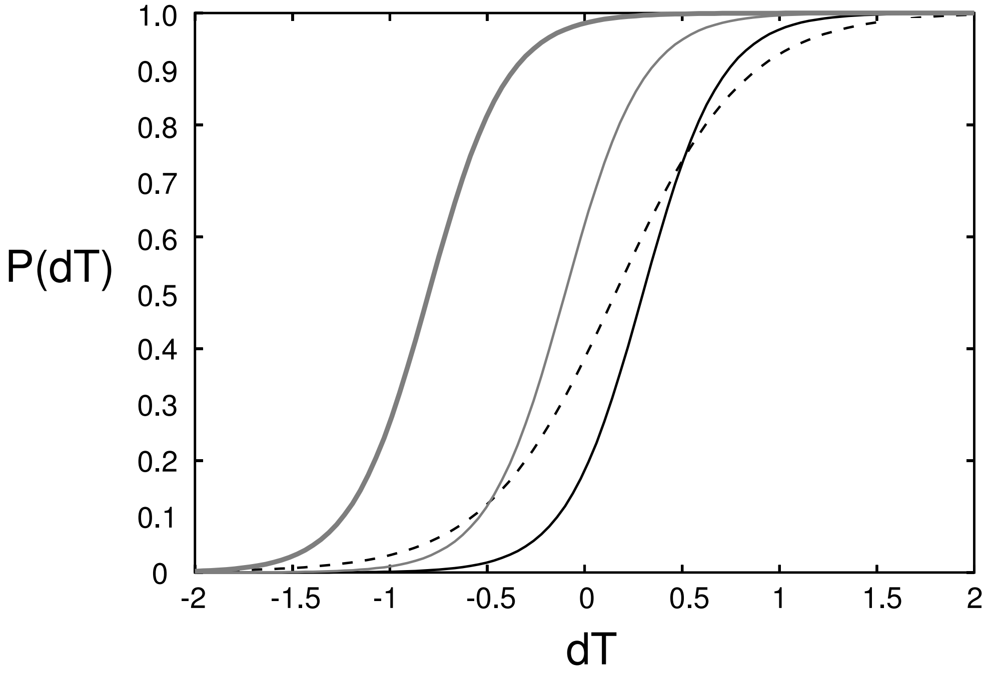
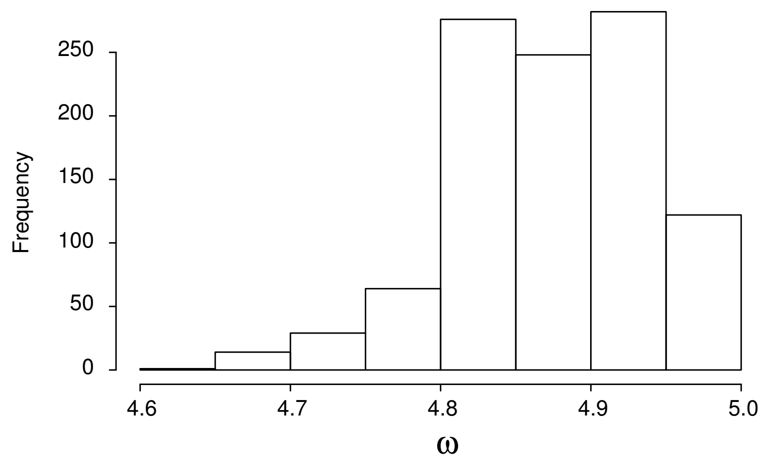


FIGURE 4.3 Examples of typical reaction norms in simulation of sexual reproduction model in the field condition. The reaction norms of uPLD (gray bold line), bPLD (gray solid line), bPFI (broken line), tPFI (solid line) are illustrated. The reaction norms of bPFI (broken line), tPFI (solid line) were the same as those in simulation of asexual reproduction model in the field condition. uPLD $(\omega, \theta) = (5.0, -4.0)$; bPLD $(\omega, \theta) = (5.0, -0.5)$.



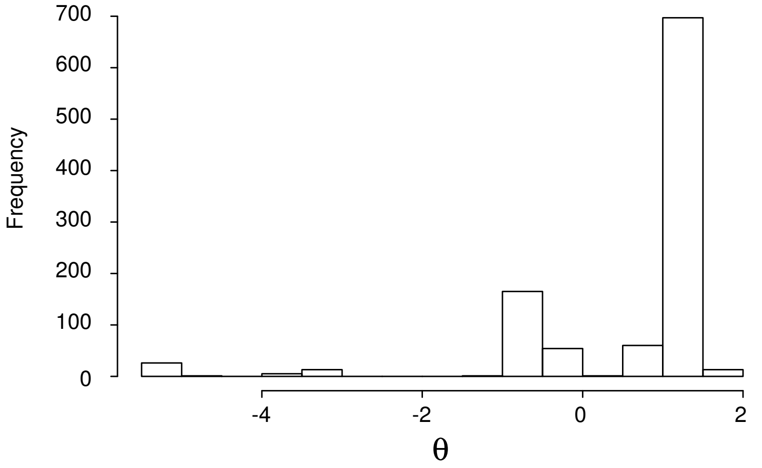


FIGURE 4.4 The distribution of ω and θ among adults on the 18000-th day in a replicate which showed large genetic variation of θ in simulation of sexual reproduction model in the field condition. This population maintained genotypic polymorphism of tPFI, bPLD, uPLD.

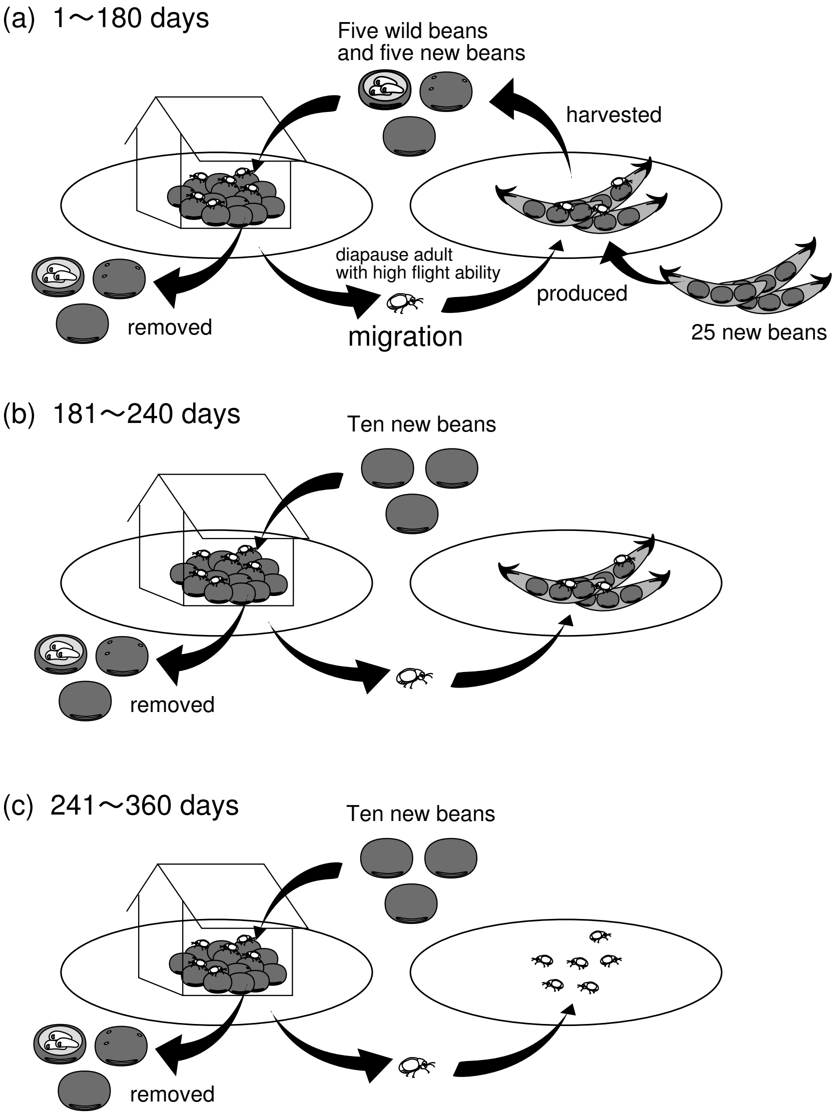


FIGURE 5.1. Resource dynamics in heterogeneous condition of field and storage. (a) harvest-time; (b) before rainy season; (c) rainy season.

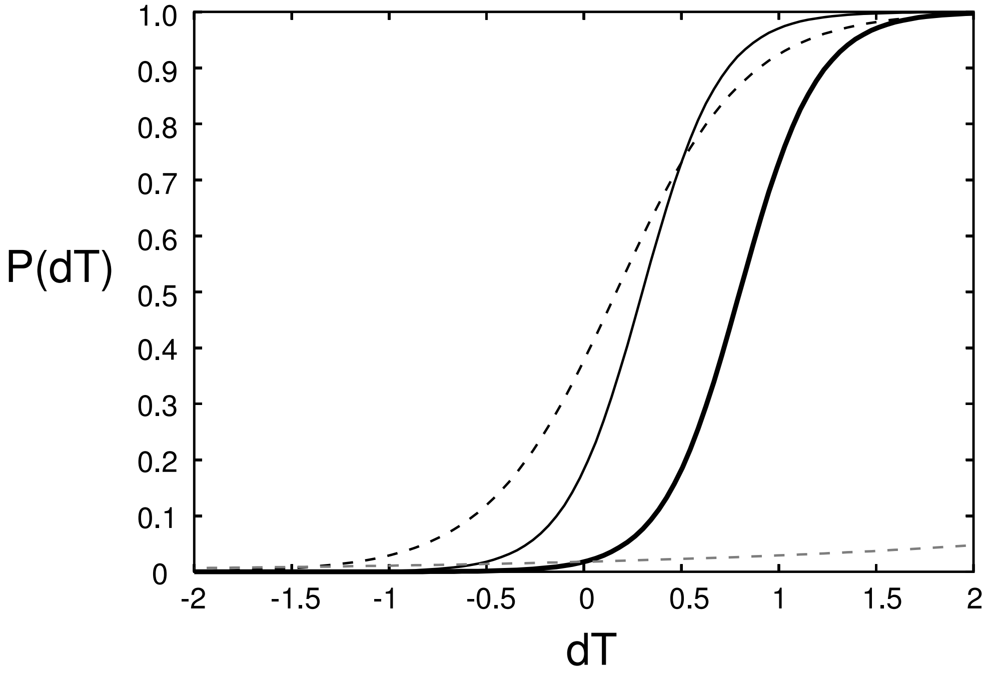


FIGURE 5.2 Examples of typical reaction norms in simulation of asexual reproduction model in heterogeneous condition of field and storage. The reaction norms of bPFI (broken line), tPFI (solid line) were the same as those in simulation of asexual reproduction model in the field condition. sPFI (bold line) and IL (gray broken line) were adaptive to the storage. sPFI $(\omega, \theta) = (5.0, 4.0)$; IL $(\omega, \theta) = (0.5, 4.0)$.

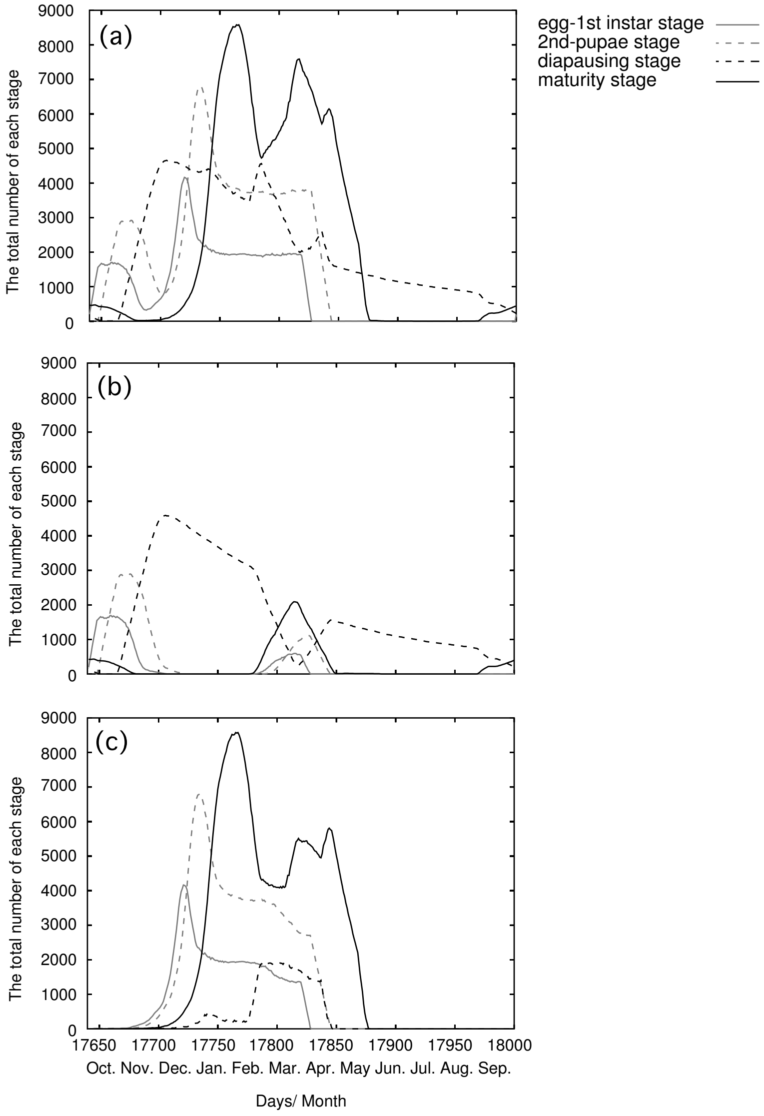


FIGURE 5.3 Population dynamics of each stage group for final year in the field in the replicate where sPFI was dominant in the storage. (a) Population dynamics of bPFI and sPFI; (b) population dynamics of bPFI; (c) population dynamics of sPFI.

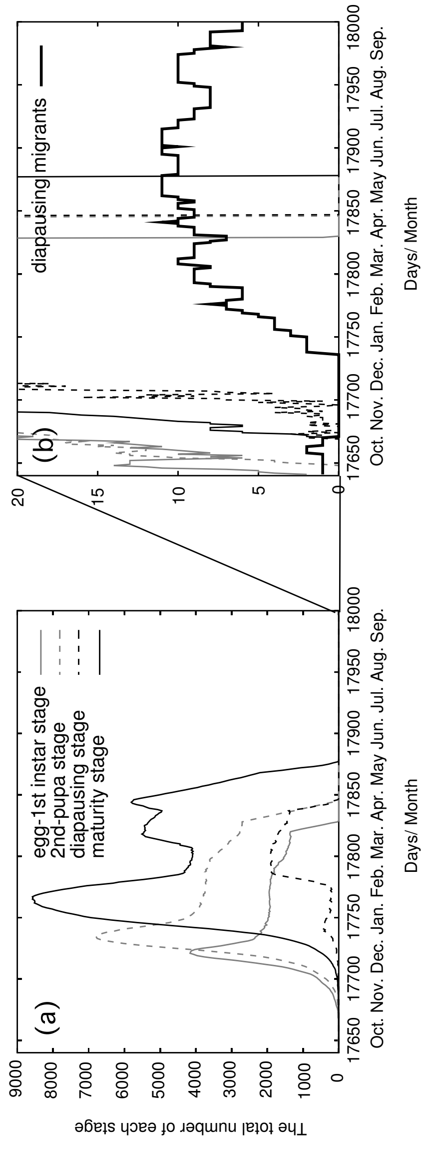


FIGURE 5.4 sPFI migrants from the storage to the field for final year in the field in the replicate where sPFI was dominant in the storage. (a) Population dynamics of sPFI; (b) sPFI migrants from the storage to the field (bold line).

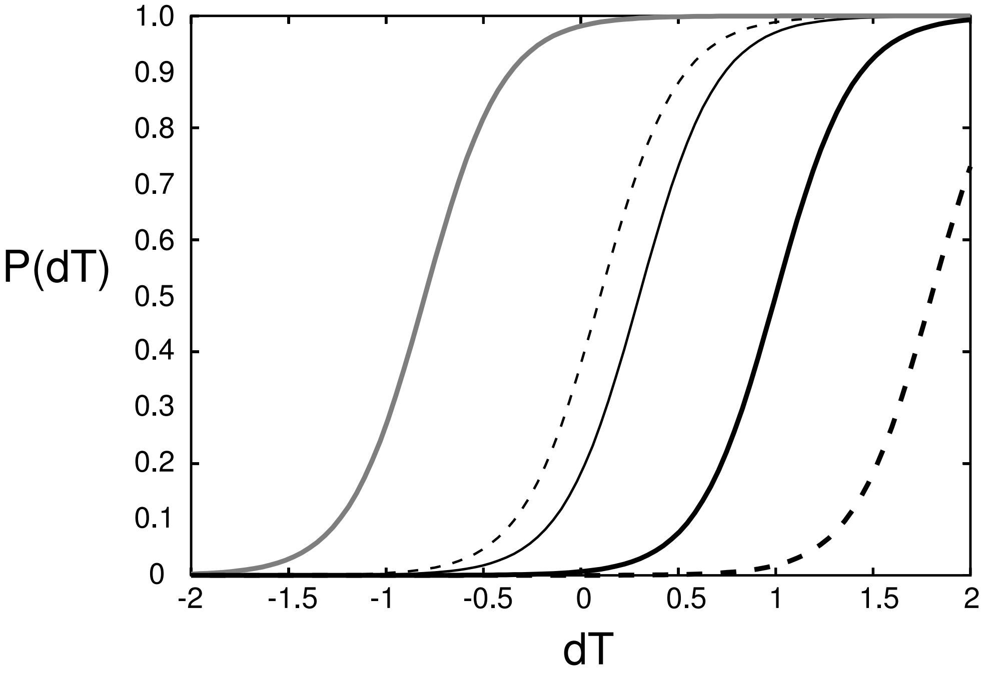


FIGURE 5.5 Examples of typical reaction norms in simulation of sexual reproduction model in heterogeneous condition of field and storage. The reaction norms of uPLD (gray bold line), bPFI (broken line) and tPFI (solid line) were observed in both field and storage. sPFI (bold line) and nPFI (bold broken line) were observed in the storage. uPLD $(\omega, \theta) = (5.0, -4.0)$; bPFI $(\omega, \theta) = (5.0, 0.5)$; tPFI $(\omega, \theta) = (5.0, 1.5)$; sPFI $(\omega, \theta) = (5.0, 5.0)$; nPFI $(\omega, \theta) = (5.0, 9.0)$.

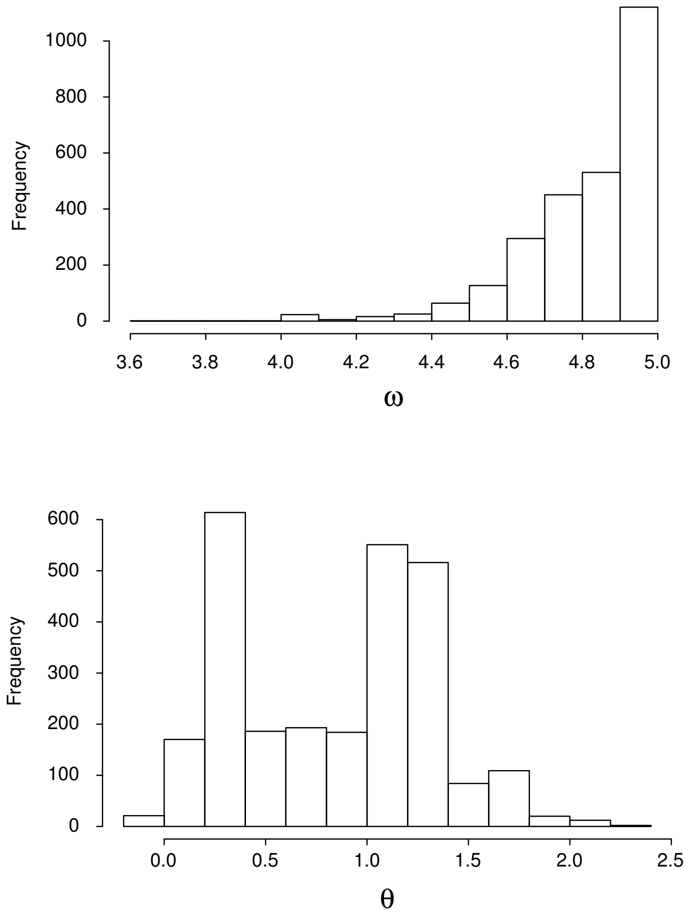


FIGURE 5.6 The distribution of ω and θ among adults in the field on the 18000-th day in simulation of sexual reproduction model in heterogeneous condition of field and storage. This population maintained genotypic polymorphism of bPFI and tPFI.

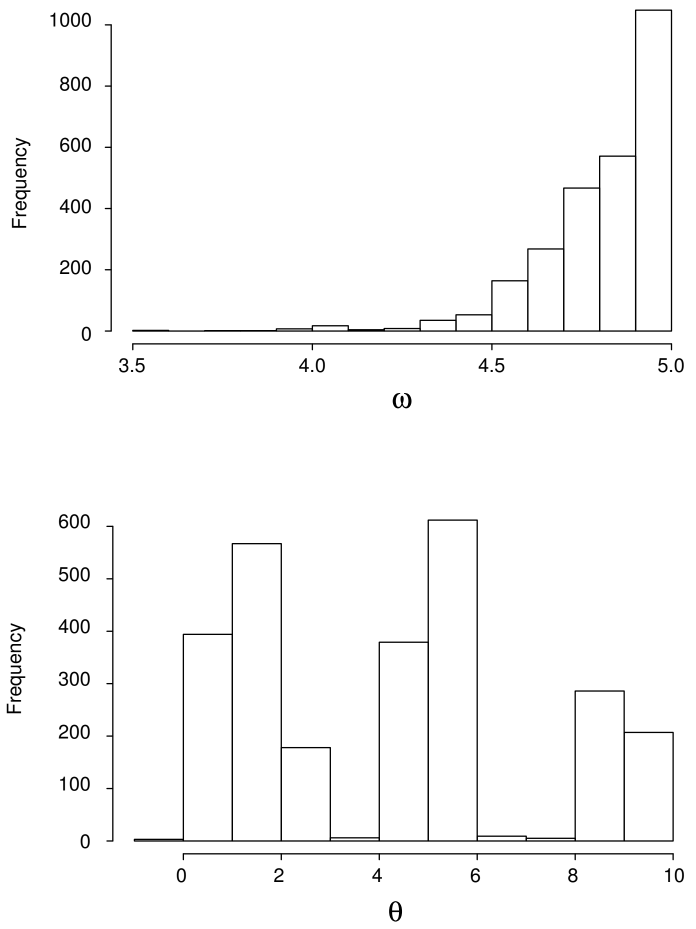


FIGURE 5.7 The distribution of ω and θ among adults in the storage on the 18000-th day in simulation of sexual reproduction model in heterogeneous condition of field and storage. This population maintained genotypic polymorphism of bPFI, tPFI, sPFI, and nPFI.

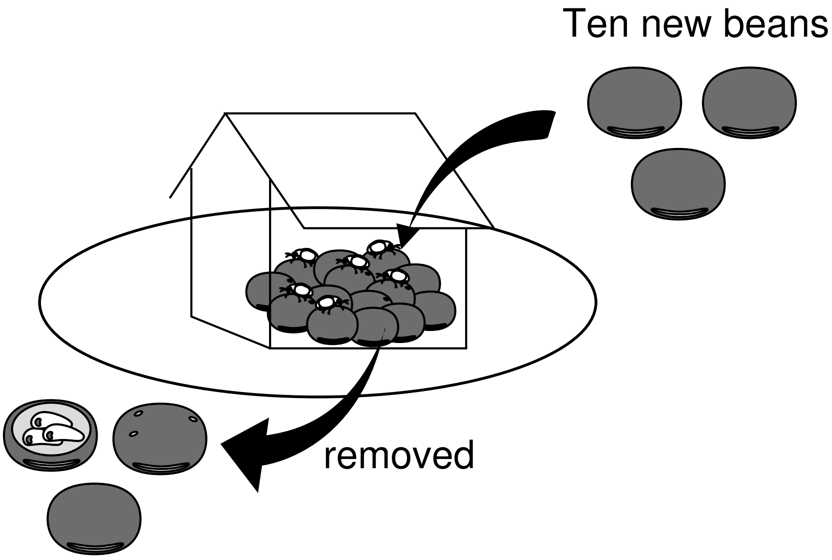


FIGURE 6.1. Resource dynamics in the storage condition.

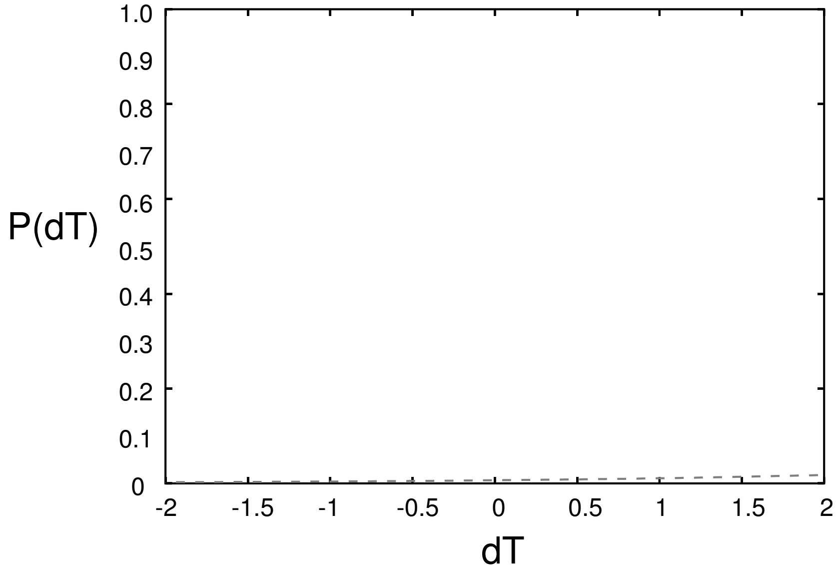


Figure 6.2 Examples of typical reaction norms in simulation of asexual reproduction model in the storage condition. IL (gray broken line) was advantageous overwhelmingly in the storage. IL $(\omega, \theta) = (0.5, 5.0)$

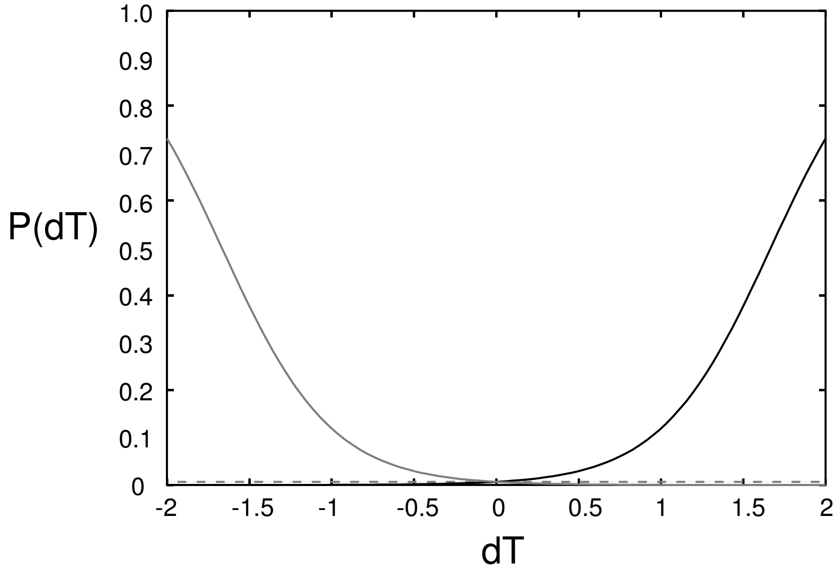


FIGURE 6.3 Examples of typical reaction norms in simulation of sexual reproduction model in the storage condition. Genotypic polymorphism of IL (gray broken line), PFI (solid line) and PFD (gray solid line) is maintained in all replicates. IL $(\omega, \theta) = (0.0, 5.0)$; PFI $(\omega, \theta) = (3.0, 5.0)$; PFD $(\omega, \theta) = (-3.0, 5.0)$

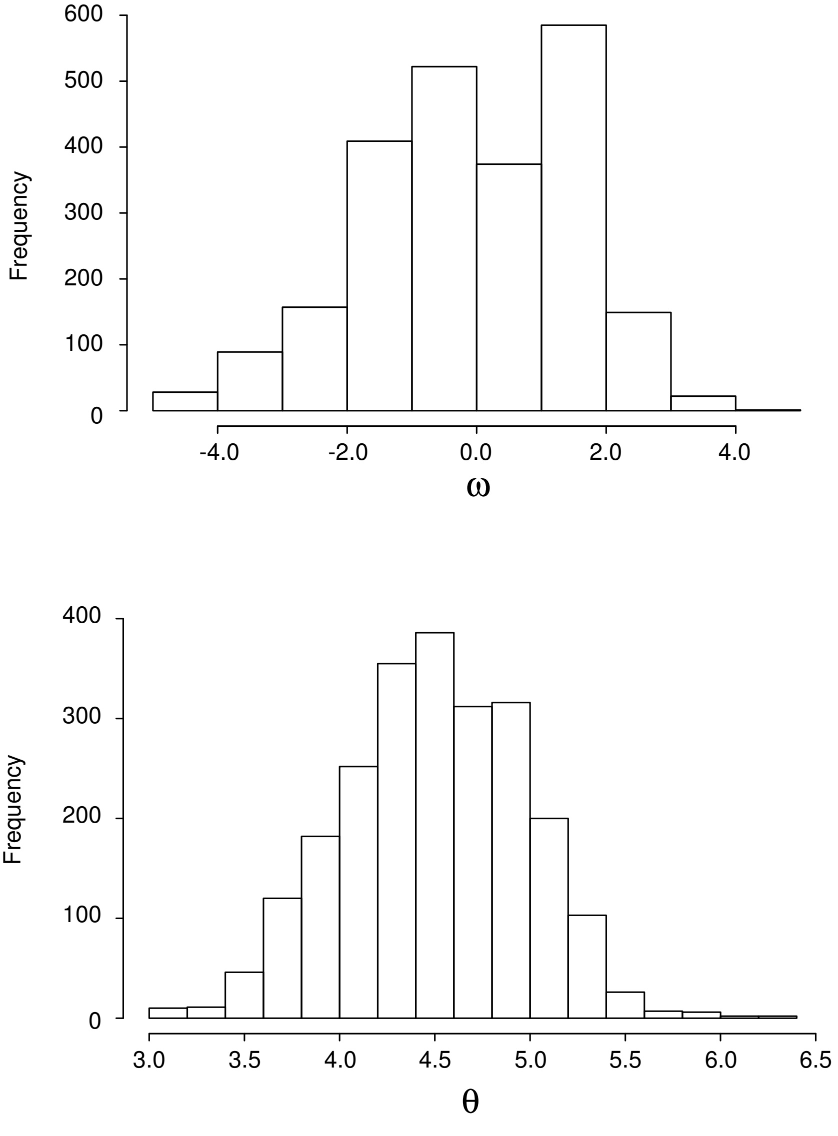


FIGURE 6.4 The distribution of ω and θ among adults on the 18000-th day in simulation of sexual reproduction model in the storage condition.