A Distributed Implementation of an Individual-Based Predator-Prey Model

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Linda Ellen Mellott

May 1997

Acknowledgments

I thank my advisor, Dr. Michael Berry, for his encouragement, support, and guidance throughout this project. I also thank Dr. Louis Gross for the opportunity to work on this project and Dr. David Straight for serving on my thesis committee.

I thank Jane Comiskey for her advice on the original SIMPDEL model and Cathy Abbott for her work on the PSIMPDEL model.

This research has been supported by the National Science Foundation under Grant No. NSF-BIR-93-18160.

Abstract

This thesis presents a distributed implementation of the **S**patially-Explicit **I**ndividual-Based Simulation **M**odel of Florida **P**anther and White-Tailed **D**eer in the **E**verglades and Big Cypress **L**andscapes (SIMPDEL) model. SIMPDEL models the impact of different water management strategies in the South Florida region on the white-tailed deer and the Florida panther populations. The SIMPDEL models the interaction of the four interrelated components, vegetation, hydrology, white-tailed deer, and Florida panther, over a 23-year period. Very similar outputs of bioenergetic and survival statistics were obtained from the serial and distributed models. A performance evaluation of the two models revealed moderate speed improvements for the distributed model (referred to as DSIMPDEL). The 4-processor configuration attained a speed improvement of 3.83 with small deer populations on an ATM-based network of SUN Ultra-2 workstations over the serial model executing on a single SUN Ultra-2 workstation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The landscape of South Florida is a complex environment that has been subjected to years of environmental stress. Disruptions in the natural water flows have been the catalyst for profound changes in the vegetation and animal life in the region. Attempts are now being made to repair the devastating effects of these changes in the water flow on the ecosystem of the South Florida region [FDG+94]. The effects of these corrections must be modeled to ensure that these new changes do not further harm the fragile region. The most effective way to evaluate the effects of changes on this complex environment is through computer modeling [FDG+94]. The **A**cross **T**rophic **L**evels **S**ystem **S**imulation for the Everglades and Big Cyprus Swamp (ATLSS) [FDG+94] family of models was developed to address modeling the impacts of environmental changes on this region. The **S**patially-Explicit **I**ndividual-Based Simulation **M**odel of Florida **P**anther and White-Tailed

Deer in the **E**verglades and Big Cypress **L**andscapes (SIMPDEL) model [CGF+96] is one component of the ATLSS model. SIMPDEL was developed to model the interaction of the vegetation, hydrology, white-tailed deer, and Florida panther in the region.

1.1 Development of Models

The serial SIMPDEL model demanded extensive computer resources in the form of memory requirements and execution time. To address these memory and time requirements, a parallel implementation was developed for the Thinking Machines CM-5 [Abb95]. The **P**arallel **S**patially-Explicit **I**ndividual-Based Simulation **M**odel of Florida **P**anther and White-Tailed **D**eer in the **E**verglades and Big Cypress **L**andscapes (PSIMPDEL) incorporated the vegetation, hydrology, and deer components of the SIMPDEL model [ABC+97]. This implementation provided an accurate reflection of the SIMPDEL model in a much-improved time. The limitations to the PSIMPDEL implementation were that the model was developed specifically for the Thinking Machines CM-5, and did not include the panther component of SIMPDEL. This need for a model that incorporated the panther component and could be executed on a network of workstations was the motivation for the development of the **D**istributed **S**patially-Explicit **I**ndividual-Based Simulation **M**odel of Florida **P**anther and White-Tailed **D**eer in the **E**verglades and Big Cypress **L**andscapes (DSIMPDEL) model. The following chapters describe the background work of the SIMPDEL and PSIMPEL implementations as well as a description of the DSIMPDEL implementation, verifications of the model and performance results.

1.2 Notation

In the following chapters, the notation $500m$ grid cell is used to refer to a grid cell that contains information about an area that is $500m \times 500m$. A $100m \times 100m$ grid cell is referred to as a $100m$ grid cell. P_i, P_j, and P_k are used to represent processor ID numbers of the machines used in the simulation. The notation P_i is always used to represent the current processor. P_j and P_k are used to represent any other processors participating in the simulation.

Chapter 2

SIMPDEL and PSIMPDEL Models

The study area for the SIMPDEL model includes 7,500 mi^2 of the southern portion of Florida. The model consists of four interrelated components that simulate the vegetation, hydrology, deer, and panther in the region. Inputs derived from historical information about the study area fuel the vegetation and hydrology components of the model. The hydrology component influences vegetative growth and limits the movement of the deer and panther. The vegetation component provides forage for the deer and influences animal movement by providing cover for both the deer and panther. The deer provide an important source of prey for the panther.

This model was developed first as the sequential SIMPDEL implementation

[CGF+96] and then parallelized for the Thinking Machines CM-5. This initial parallel attempt, PSIMPDEL, incorporated the vegetation, hydrology, and deer components of SIMPDEL [Abb95]. This Chapter describes both background work in PSIMPDEL and the sequential implementation of the panther component.

2.1 Landscape

The study area for the SIMPDEL and PSIMPDEL models encompasses a region of approximately 7,500 mi^2 in South Florida. The climate in this region is subtropical, which gives the area distinct wet and dry seasons. The wet season occurs from May to October, with approximately 80% of the recorded precipitation falling during this period [SB94]. The landscape consists of eighteen distinct vegetation types, the two most prevalent being freshwater marsh and forested wetlands. This region is essentially flat, with elevations that range from 10m in the northern portion of the study area to 1m in the south [CGF+96].

The magnitude of the study area makes it appropriate to model the landscape in maps of two different spatial scales: 100m grid cells and 500m grid cells. Each of the 500m grid cells is comprised of 25 100m grid cells. Both the 500m and 100m grid cells contain information about the vegetation, hydrology, and forage levels of the area. The information in these maps is updated by the vegetation and hydrology components of the SIMPDEL model at regular intervals. The deer and panther components of the model rely upon the maps for information about vegetation and water levels [Abb95].

Hydrology Component

The hydrology data provides information about the region in Southern Florida that is bounded by Lake Okeechobee in the north and extends south to Florida Bay, an area of approximately 7,500 mi^2. This region includes part of the Big Cypress National Preserve, Everglades National Park, and Water Conservation Areas 1, 2, and 3. The hydrology data is derived from the South Florida Water Management Model developed by the South Florida Water Management District and the Everglades National Park. This hydrology model uses the elevation and historical rainfall data from the period from 1966 to 1989 and calculates daily, and averaged weekly and monthly hydrology data [CGF+96]. The original scale of these outputs was at the $2mi$ scale. This was incompatible with the vegetation data, which is at a much finer resolution. Spatial data interpolation was used to redistribute the water levels to represent the values in each of the $100m$ cells of the vegetation data. This redistribution was based upon the expected water depths for the vegetation types in each $2mi$ cell [CGF+96].

Vegetation Component

The vegetation data was derived from the Florida Department of Transportation vegetation map. The map is a satellite image that represents 10,000 mi^2 of southern Florida at a $30m$ resolution. This map is larger than the hydrology data, and currently only the portion corresponding to the available hydrology data is used in the SIMPDEL model. The map provides information at both the $500m$ and $100m$ scale [FDG$^+$94] about three different qualities of deer forage for inputs to the deer model, and information about ground cover for both the deer and panther models.

The deer rely upon the vegetation component for forage. During a simulated day a $500m$ grid cell is selected for each deer during the foraging process. Then one of the 25 $100m$ pixels from within that grid cell is selected and the deer removes forage at that level. The vegetation data can be updated at daily, weekly, or monthly intervals, producing the effects of seasonal vegetation growth. The vegetation data provides three distinct classes of biomass that are differentiated by caloric content and quantity. High quality forage has the highest caloric content, $1800kcal/kg$, but is the lowest in quantity. Medium quality forage provides a lower caloric value, $1200kcal/kg$, than high quality forage, but is more plentiful. Low quality forage is provided in an unlimited quantity, but provides the least caloric value, only $800kcal/kg$. The deer place a priority on high quality forage, and will eat from the other forage levels only when high quality forage is not available [CGF+96] [Abb95].

Landscape Partitioning

The landscape in the PSIMPDEL model was partitioned across the 32 processors of the Thinking Machines CM-5. The partitioning results in each processor containing only a portion of the landscape, reducing the number of calculations needed for vegetation and hydrology updates and greatly reducing computation time. A row-wise block-striped partitioning strategy method of partitioning was used in PSIMPDEL [Abb95]. This method allowed for each processor to own approximately the same number of $500m$ grid cells, although the number of rows owned might vary. This difference in the number of rows owned by a processor can be attributed to the irregular study area, resulting in some some rows containing more grid cells than others. This partitioning results in each processor P_i, having two nearest neighbors. P_{i-1} to the north and P_{i+1} to the south. This is true for all processors except P_0, which has no northern nearest-neighbor, and P_{max}, which has no nearest-neighbor to the south [Abb95].

2.2 Deer Component

The white-tailed deer are the largest native herbivore in the study area. Inputs from the vegetation and hydrology components provide information about food and ground cover to the deer component. The deer population is an important source of prey for the panther represented in the simulation.

2.2.1 Sequential Implementation

The deer component of the SIMPDEL model simulates the activities of the deer population on a daily time-step for a 23 year simulation. Each deer is represented as a structure that provides information about the individual states of that animal. These structures are created at the beginning of each simulation and represent characteristics such as age, weight, sex, and location [CGF+96].

The deer component contains six activities: aging, dispersal, growth, mortality, reproduction and foraging. The age of each deer is maintained as an individual characteristic of that animal and is incremented on a monthly basis. Dependent young travel and forage with their mother until dispersed from the natal range. The simulation of the reproduction process allows a monthly search by adult females for a male before initiating a pregnancy. All animals forage on a daily basis. During the forage simulation the $500m$ grid cells surrounding the current location of the deer are searched for suitable forage. Once a grid cell is chosen, one of the 25 $100m$ grid cells within the $500m$ grid cell is selected and the deer removes vegetation from the $100m$ grid cell. The quality and quantity of available forage affects the growth of the deer. Deer growth is calculated by the subtracting the daily energy expenditures from travel, grazing, and reproduction from the energy intake from foraging. If the energy gained from foraging is greater than the total energy expenditures, the deer will gain weight, otherwise the deer will lose weight. Deer mortality occurs when the weight of an animal drops below 70% of the maximum weight ever attained by the animal. Other deer mortality factors include natural mortality, disease, and predation by panther [CGF+96] [ABC+97] [FDG+94].

2.2.2 PSIMPEL Implementation

The deer model was first parallelized in PSIMPDEL on the Thinking Machines CM-5. The landscape in this model was partitioned across the 32 processors of the CM-5 as described above. The deer population was also divided among the processors, so that each processor contained information only about the animals located in the grid cells owned by that processor. Restricting animal movement to the current processor lead to inconsistencies in the model, so a message passing system was introduced. The messages allowed the deer to utilize the entire search distance when dispersing, reproducing, and foraging [Abb95] [ABC+97].

2.3 Sequential Panther Component

The Florida panther once roamed the region encompassing the Gulf and Atlantic coastal plains of the southeastern United States. Now the animals are limited to a small area in southern Florida [SB94]. The panther habitat is limited to the predominantly forested areas in the region. The panther feed primarily on the white-tailed deer and feral hogs that are found in the region [Bel89].

The panther component of the SIMPDEL model contains six main components: aging, reproduction, predation, dispersal, growth, and mortality. Each panther is represented as a structure that is created and initialized with individual characteristics at the beginning of each simulation. These individual values determine each panther's response to the different rules that govern panther behavior in each portion of the model. The values are then updated to reflect the outcome of those behavorial rules.

The panther component progresses on a daily time-step over the 23 years of the simulation, the same time-step as the deer component. The panther model interacts with the deer, vegetation, and hydrology components of SIMPDEL. The vegetation type and the water level restrict panther movement throughout the habitat. The deer model provides the panther with an important source of prey.

2.3.1 Reproduction

The reproduction process is simulated only for those panther that have attained the minimum mating age for their gender. For males, this minimum age is 24 months. Mature male panthers are available for mating when they are not at a location where prey was taken and are not currently serving a female. A mating search to locate available females is conducted for all males who satisfy these criteria and have an individual mating probability greater than the general male mating movement probability when 30 days pass without mating [CGF+96].

The male panther mating search is conducted by iterating through the array of panther and evaluating the distance from the panther for which the mating search is being conducted to each available female. The search may extend to the male mating distance of 100 $500m$ grid cells. Locating a female within that range allows the male to maintain its current location. If no available females are observed within the mating distance, the male is relocated to a location that is closer to the nearest available female identified during the mating search.

The minimum mating age for a female panther is 18 months. Each month on a female panther's mating day several factors are evaluated to determine if a search for a mate is required. The preconditions to mating are: the panther must not be pregnant, must not have any dependent kittens of age less than 18 months, and must be healthy. The state of the female's health is determined by evaluating the female's current weight and determining if it is greater than the mean weight for her age [CGF+96].

Satisfaction of the preconditions listed above allows a female to search for an eligible male within the mating distance. The search pattern for a female panther is an expanding square centered on the grid cell location of that panther. The maximum depth of this search is 100 $500m$ grid cells. From the eligible males that are located, a mate is selected based on an age priority that gives preference toward older mates. The selected male is marked as mating for the maximum number of mating days, currently seven, making it unavailable for predation and for mating with other females during that period. If the conditions for mating are not satisfied, the mating day for the female is incremented and the search is repeated each month until a pregnancy is initiated.

The gestation period for the panther is 90 days. Failure of a pregnancy results from poor health or weight loss in the expectant mother. New mothers must be independent with no dependent kittens, and if dispersal is necessary it is conducted prior to the female giving birth. The maximum litter size is six kittens, while the actual litter size for each mother is determined by litter size probabilities and the ratio of the expectant mother's body weight to the mean body weight for her age class [CGF+96].

2.3.2 Predation

The three sources of prey for the panther are deer, panther, and feral hogs. The deer are simulated in the deer component of the model and the feral hogs are represented in the *hog map*, a two-dimensional array. Each entry in the hog map corresponds to a grid cell in the landscape map. The base hog map is initialized from the input files at the beginning of each simulation. The hog map is depleted as the hogs are chosen as prey. Twice a year the hog map is replenished to simulate the death and rebirth cycle of the natural population. Not depleting the hog map leads to an overkill of hogs and insufficient predation of the deer population.

The panther predation search is at the $500m$ scale. Young panther remain with their mother until dispersed from the natal range. Adults will not search for prey while at a grid cell where prey was taken or during the mating period. The predation process begins in the current grid cell. A grid cell must have a minimum number of deer to guarantee a deer kill. If predation does not occur in the current location, the predation search expands to the surrounding grid cells. The expanded search continues until either a deer, panther, or hog is selected as prey, the panther is displaced from the search area or is killed by a dominant panther, or the panther's maximum travel distance for the day is exceeded. The expanded search begins with an examination of the surrounding eight grid cells for one with both a suitable habitat and a population of deer (see Figure 2.1). After locating such a grid cell, then a kill may be made from any of the deer positioned there. If no prey are taken, but deer are present in one of the grid cells searched, the same eight grid cells are searched again, and this time the grid cell with the highest concentration of deer is identified. Absence of deer in any of the grid cells calls for a another search of the eight grid cell area, and here the grid cell with the highest concentration of hogs is located. The grid cell selected during the predation search is then checked for the presence of another panther. If more than one panther is present in a grid cell the individual characteristics of each animal are evaluated, and the weaker panther is either killed or displaced

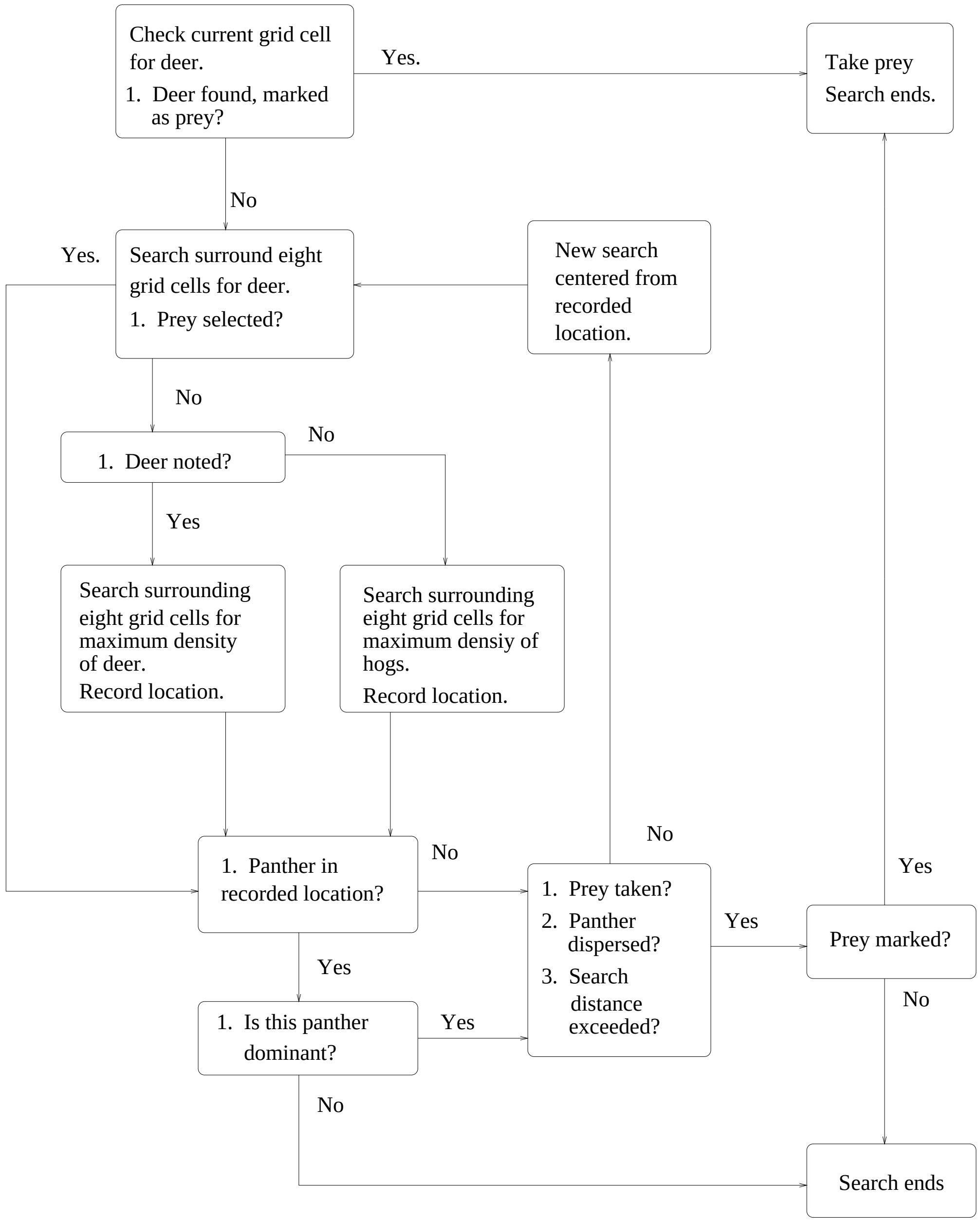

Figure 2.1: Serial Predation Flow

from the grid cell by the dominant panther.

If the panther is alive and is not displaced at the end of the searching process a kill is made on the selected prey. If no prey is selected during the searching process, the maximum hog density identified during the predation search is evaluated. If the density is great enough to ensure predation, the panther is relocated to the identified grid cell and a hog is removed from the hog map. The number of days without a kill is recorded and after fourteen days without a kill, the panther attempts a short range dispersal. Twenty-one days without a kill prompt a long range dispersal. The dispersal process is described in Section 2.3.5.

2.3.3 Growth

The factors that determine panther growth are size and frequency of kills, daily travel distance, and for females, pregnancy or dependent young. These factors are evaluated on a daily basis to obtain an energy balance that determines the new weight of the panther.

The daily energy balance for each animal is calculated by subtracting the energy expenditures due to basal metabolism, travel, and predation from the energy gains that are made from food intake. More energy is required by females who are pregnant or have dependent young under the age of three months and by males who are mating. Without deer and hog kills, other sources of prey supply full energy for a panther 40 percent of the time, while less than full energy is gained the remaining 60 percent of the time. A positive energy balance results in a weight increase, while a negative balance results in a decrease in weight [CGF+96].

Young panther maintain their weight unless their mother has a weight loss or gain, then the young panther's weight is recalculated. Kittens under the age of three months grow at the predetermined growth rate for their gender. After that age, weight change is determined by calculating the daily energy expenditures of the panther minus the energy obtained from prey.

2.3.4 Mortality

The adult panther mortality factors are weight loss, panther aggression, disease, natural mortality and dispersal. In addition, deaths of dependent young result when their parent dies while the young panther is under the minimum age for independent survival. The probability of death by disease is dependent upon the season of the year. Natural mortality rates vary by age and increase with the density of the panther population. Deaths due to weight loss occur when a panther's weight drops below two-thirds of the maximum weight attained. Panther are territorial animals, and deaths from panther aggression are caused when an animal is moved into a grid cell location that contains a dominant animal during the predation process. Dispersal deaths occur when a large number of attempts are made to find a new location. This implies long travel to the dispersal location and increases the panther's vulnerability to mortality factors.

2.3.5 Dispersal

The dispersal function simulates the dispersal of mature offspring from the natal range, displacements due to lack of prey, and movement necessary to avoid other panthers. The process begins by generating a random grid cell location. That location is then checked for both a suitable habitat and to ensure that it is within the dispersal distance. The habitat must provide an appropriate water level for the age or gender of the animal. The vegetation type must be from one of five vegetation types acceptable to panther. Once a grid cell with an appropriate water level and vegetation type is located, the distance to the grid cell from the panther is evaluated. If the distance is less than the dispersal limit for the given type of dispersal, then the grid cell is selected and the panther is relocated.

Short range dispersal is limited to the maximum travel distance for that panther. This type of displacement is used leaving the natal range, or after 14 days without locating a prey. In addition, females will make a short range dispersal when encountering another panther during predation. The long range dispersal encompasses an area of twice the maximum travel distance and is conducted after 21 days without a kill, or when males encounter a dominant animal during predation [CGF+96].

Chapter 3

Parallel Implementation

The parallel distributed implementation of the SIMPDEL model uses MPI message passing functions. Changes were made to the message passing in the vegetation, hydrology, and deer components to replace the CMMD message passing functions with appropriate MPI commands. These changes and the modifications made to parallelize the panther component are described in the following sections.

3.1 Implementation Details

The MPI protocol provides functions for the sending of messages consisting of homogeneous, contiguous data. The sending of heterogeneous or noncontiguous data requires special processing. Each element in a message must be packed into the message buffer prior to sending. This packing can be done with the MPI_Pack function or through the use of derived datatypes. MPI_Pack is recommended for messages that are sent a limited number of times because there is an additional cost incurred for each invocation of the function. Derived datatypes incur an additional overhead when the datatypes are formed, but less for each individual send [SOHL+96]. Due to the potentially large number of heterogeneous messages sent in each simulation, derived datatypes are calculated for each of the message types during the setup of the simulation. This is done by first defining the message structures, calculating the address offset of each message field, and then creating a new message type. These derived message types are then available for use throughout the DSIMPDEL model.

All of the receives in the DSIMPDEL model are implemented using the blocking MPI_Recv command. This is appropriate because processes probe for incoming messages and do not make a call to the receive function until the message has arrived. Sends are implemented with MPI_Isend, a nonblocking, non-buffered send command, and MPI_Ibsend, a nonblocking, buffered send. These commands take the place of the CMMD message passing functions utilized in the CM-5 implementation of PSIMPDEL [Abb95] [ABC+97].

Both the deer and panther components require the explicit synchronization of processes during a simulated day. In PSIMPDEL, this is accomplished with CMMD functions. Equivalent MPI functions are not available, so this process was modified in DSIMPDEL. The synchronization changes are discussed in Section 3.5.

3.2 Landscape

The vegetation and hydrology input data for this implementation is the same as that used in PSIMPDEL. The only required changes involved the message passing needed to repartition the map. The same map structure and row-slice landscape partitioning that was used in PSIMPDEL [Abb95] was also used in this work. However, changes were made to the partitioning process to allow the landscape to be partitioned among a variable number of processors.

3.3 Deer Component

The changes to the deer component were primarily to the message passing functions. All functions were converted from the CMMD message passing library to the appropriate MPI message passing routines. The deer structure and program flow were unchanged.

Minor changes were required to facilitate the predation process of the panther component. The deer in PSIMPDEL are maintained in an array structure. Accessing an animal at a specific grid cell required iterating through the entire deer array and evaluating the row and column values of each animal. This was an appropriate choice for PSIMPDEL because locating another deer at a specific grid cell is required only during the reproduction process. In the panther component of DSIMPDEL, the panther must also search for deer because they provide an important source of prey. This searching process required a method of associating deer with their current grid cell. A matrix of the grid coordinates corresponding to the rows of the map owned by each processor serves this purpose. The map is updated with each deer move and then accessed when panther make predation decisions, eliminating the need for extensive iteration through the deer array.

The synchronization process was also modified due to the unavailability of MPI functions corresponding to the CMMD_functions used in PSIMPDEL. These changes are addressed in Section 3.5.

3.4 Panther Component

The reproduction, dispersal, and predation portions of the panther component required modifications to the serial SIMPDEL model. Each of these components allow each processor to conduct a search of the surrounding grid cells which may extend beyond the map boundaries of that processor. The growth and mortality portions of the SIMPDEL model do not require any animal movement so these sections remained fundamentally unchanged.

3.4.1 Reproduction

The reproduction portion of the panther component requires each processor to conduct mating searches for both male and female panther on surrounding $500m$ grid cells. The parallel implementation of the reproduction process required modifications to accommodate the searches that extend across the map boundaries of a processor.

Male Panther Role in Reproduction

The reproduction process begins by initiating a search for each male panther that has not been selected for mating for 30 days. The flow of the reproduction activities conducted for male panther is shown in Figure 3.1. Initially, processor P_i performs a reproduction search for a male by iterating through the panther array and evaluating the distance from the grid cell of the male panther to the grid cell of each female in the population. The search is successful if an eligible female is located within half the male mating distance of 100 grid cells. This initial search requires no message passing and the male remains in its current grid cell location.

When the search on P_i is unsuccessful, a message is sent to initiate a search on another processor within the mating distance. The processor ID numbers of the processors within the mating distance are calculated at the beginning of each simulation and are recorded in the *mate array.* To extend the search, P_i randomly selects a processor, P_j, from the mate array and sends a reproduction message to it. Upon receipt of the mate message, P_j must perform a search to determine if its panther population contains a potential mate for the male. If such a female exists, this information is returned to P_i. Search failure results in P_j randomly

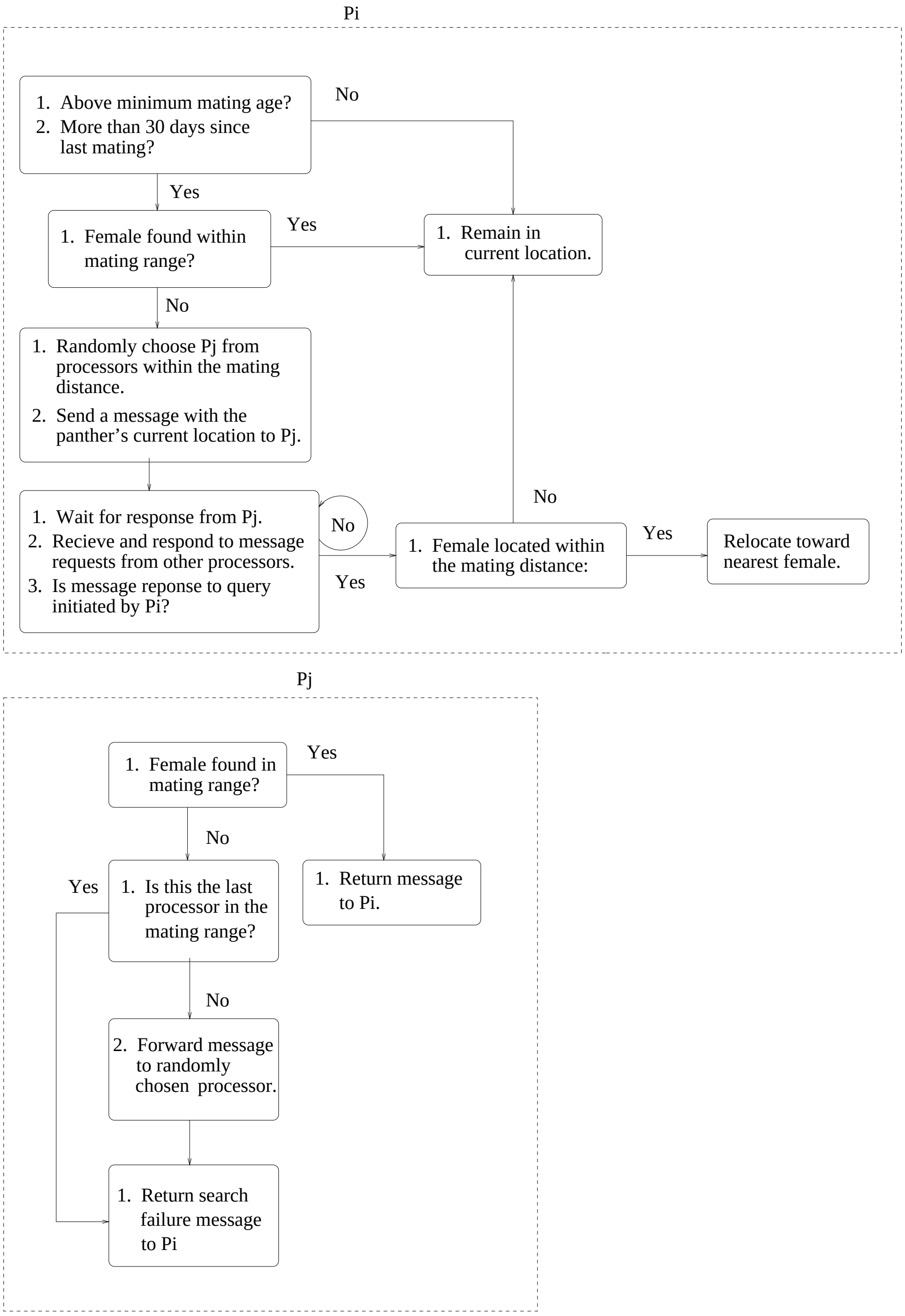

Figure 3.1: Male Reproduction Pattern

choosing another processor which has not conducted a mating search for this male and forwarding the mate message to that processor. The search continues in this manner until either an eligible female is located, or all of the processors within the male panther mating distance conduct a search. When search failure occurs on all processors in the mating distance, the last processor chosen from the mate array to perform a mating search returns a message to P_i indicating search failure.

When P_i receives the message containing the results of the extended reproduction search, P_i must evaluate the search results and determine if this male should be relocated. Unsuccessful searches require relocation of the male to a position that is closer to the nearest female on P_i. A successful search on any processor within the mating distance allows the male to maintain its current location.

Female Panther Role in Reproduction

Processor P_i must conduct a mating search to locate available males for all females eligible for mating. (see Figure 3.2). The search originates in the current grid cell and expands outward in squares of increasing size that are centered at the female's current grid cell. Searching continues until the distance covered exceeds the maximum search range of 100 grid grid cells.

If the search expands beyond the map boundaries of P_i as shown in Figure 3.3, a search message is sent to P_j, the processor containing the portion of the map that is to the north or south of the panther's location, whichever is indicated by the

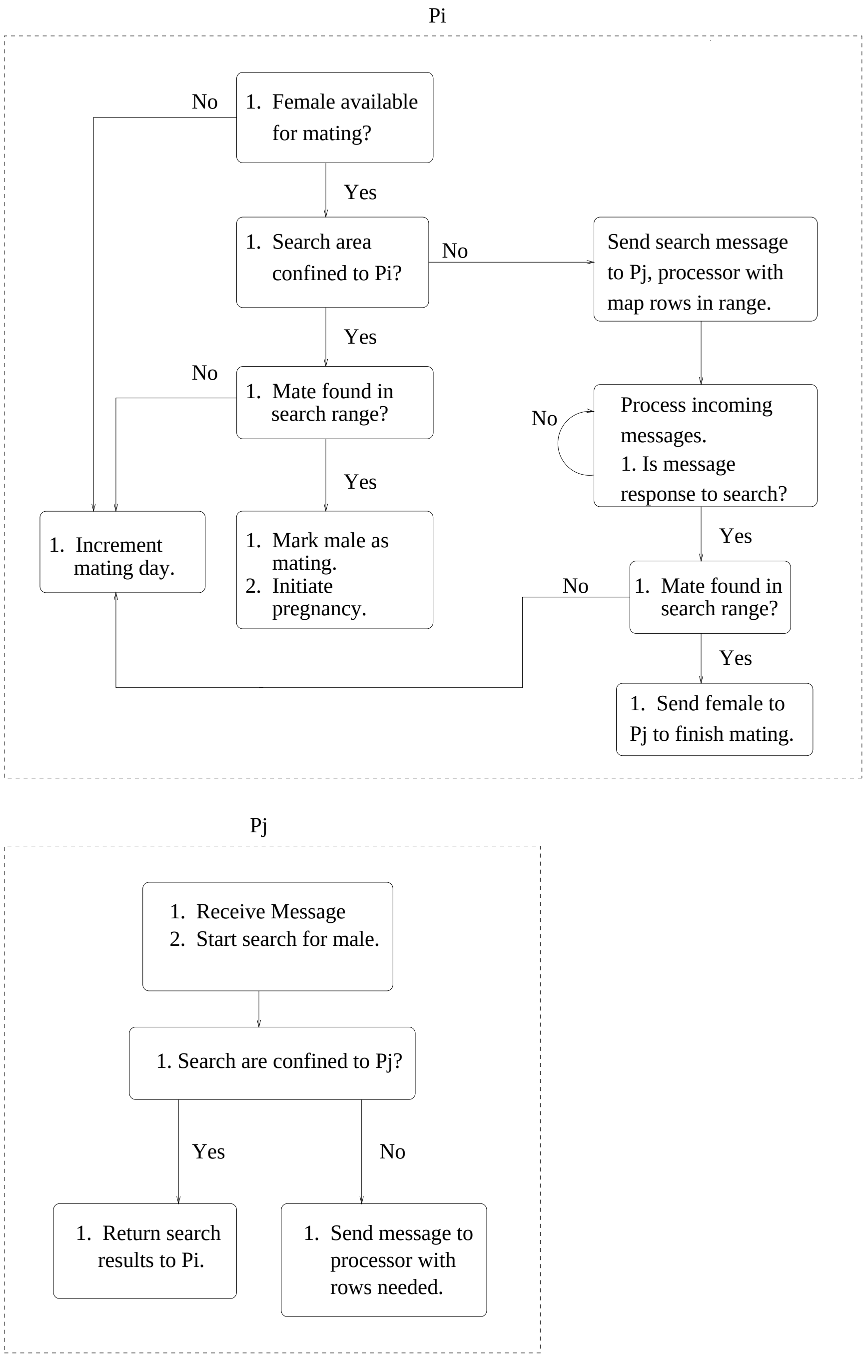

Figure 3.2: Parallel Female Reproduction Flow

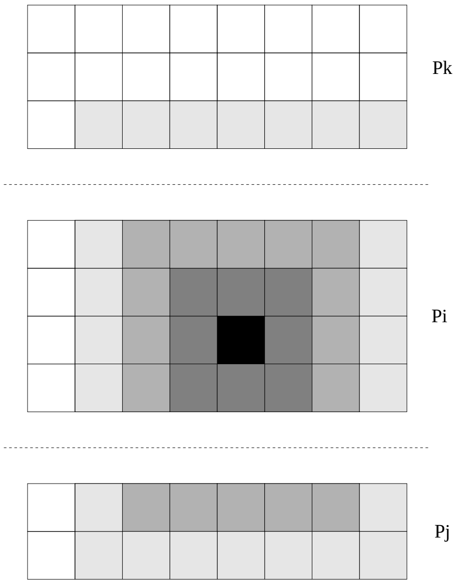

Figure 3.3: Parallel Female Reproduction Search

search pattern. Processor P_j receives this message and searches its portion of the map for available males. P_j returns information about the both the availability of males and if the search extended beyond its map boundaries to another processor. P_i waits until the messages from all processors participating in the search are received. During the search process a minimum number of males must be identified to insure conception for a female. If the minimum number is obtained, then a pregnancy is initiated for the female and the most eligible male is marked as mating so that it is not available to mate with other females during the mating period. If the eligible male is on a different processor, P_j, the female is sent to P_j where the mating process is repeated. This is necessary because the male that is the intended mate may mate with another female while P_i processes the mating messages. All available males cannot be marked as mating pending a decision on which male will be selected, as this would interrupt the mating process for other females. Such a restriction would significantly reduce the number of mating attempts and result in a reduction in the panther population.

3.4.2 Predation

All adult panther who are not at a *kill site* and not mating participate in predation. Kill sites are grid cell locations where prey was selected and which is still serving as a food source for that panther. As in the SIMPDEL model, the predation process continues until either a kill is made, the panther is displaced from the search area by a dominant panther, the panther is prey to another panther, or the maximum travel distance for the day is exceeded. The sources of prey are deer, feral hogs, and other panther. As in the SIMPDEL model, a priority is placed on deer predation, panther serve as prey only when encountered during the predation process, and the lowest priority is placed on the capture of feral hogs. Feral hogs are selected only when no other source of prey is available [CGF+96].

The predation process begins for each panther with a search for deer in the current grid cell. This requires no communication, as the current grid cell is always on P_i. When predation is not successful in the current grid cell, the search pattern is expanded to include the surrounding grid cells. When the search pattern extends beyond the map borders, search messages are sent to the appropriate processors. Restricting the panther predation range to P_i dramatically impacts the number of deer kills. This has the effect of both permitting the deer to achieve inflated population levels and increasing the number of hog kills made by the panther population.

Local Predation Searches

Panther eligible to search for prey are selected randomly on a daily basis to carry out a search. The flow of the basic predation search process is shown in Figure 3.4. First, the current location is examined for potential prey. If predation fails in the current grid cell, it is marked as the *search center*, and an expanded search of the surrounding grid cells is required. The search center is the grid cell from which the search of the surrounding eight grid cells is initiated. A new search center is selected each time the surrounding eight grid cells are examined for prey. First, the eight grid cells encompassing the current grid cell location of the panther are searched. If all of the surrounding eight grid cells are on the current processor, the number of deer in each grid cell is evaluated. A deer is selected as prey if the number of deer in the grid cell is greater than the number required to insure predation and that grid cell is marked as the new search center. The maximum number of deer noticed in a grid cell is also recorded. The grid cell with the maximum concentration of deer will become the new search center for

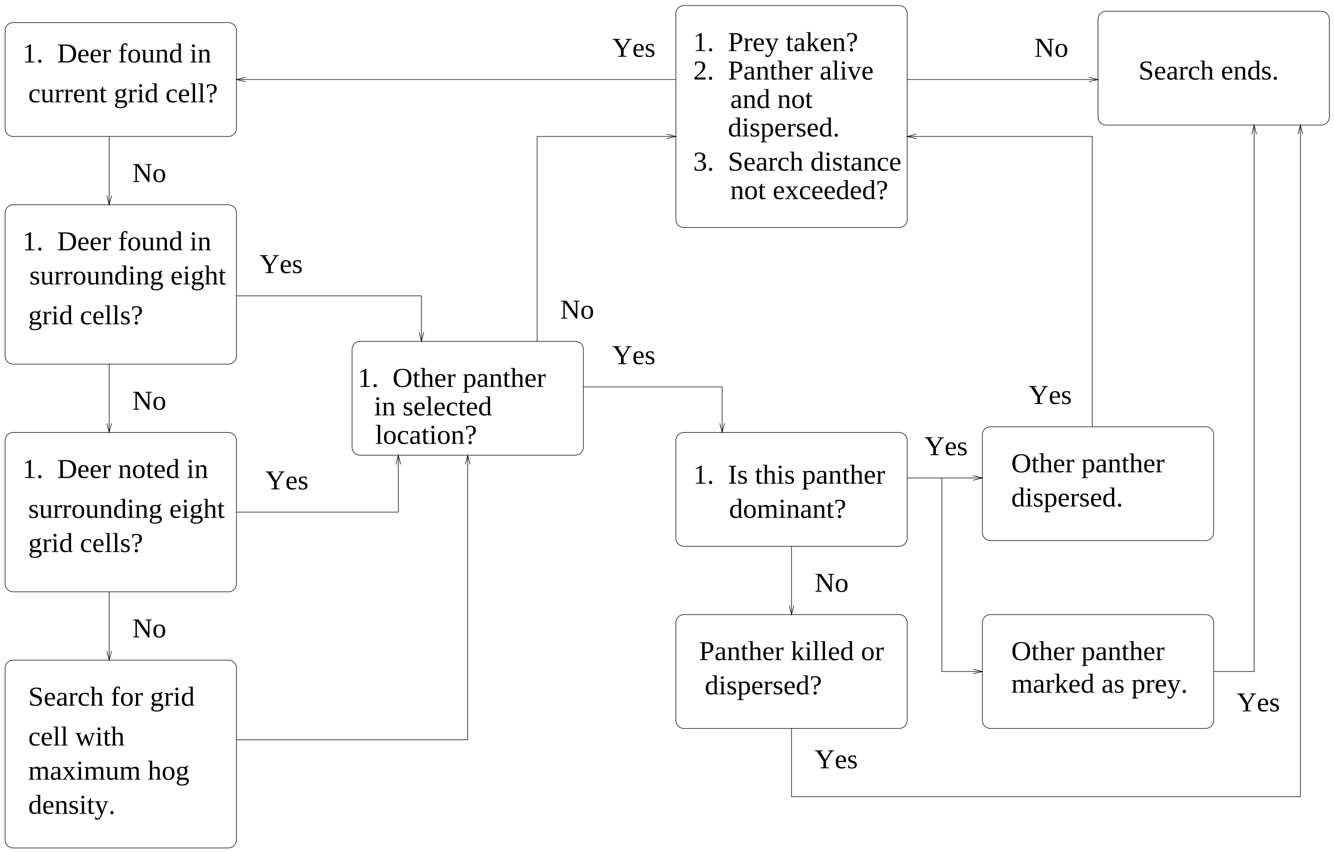

Figure 3.4: Basic Predation Flow

the predation search if no prey are taken. When no deer are marked for predation or sighted during the search, the same eight grid cells are evaluated again and the cell with the maximum concentration of hogs is selected as the search center. The grid cell selected as the search center is now evaluated for the presence of other panther. If another panther is in the search center, the characteristics of the two animals are evaluated. One of panthers must be relocated, or become a victim of aggression from the other panther. The action of this function is recorded, but is not completed until the search process is finished for the day. An array of panther encountered during each day is maintained so that each animal is encountered only once during the search process.

Checking the search center for panther is the last step in the search of the surrounding eight cells. If no prey were taken, the panther is alive and not dispersed, the search process continues for that panther until the maximum travel distance is attained. If deer were noted during the previous search, the search center from which the new search is centered is the grid cell with the highest concentration of deer. If no such pixel exists, the predation search continues from the grid cell with the highest concentration of hogs. Although the search center changes each time the surrounding eight grid cells are examined, the grid cell location of the panther is not updated until the entire search is concluded.

Expanded predation searches

If any of the eight grid cells surrounding the search center are beyond the map boundaries of processor P_i an expanded predation search is required. First, P_i sends a predation message to P_j, the processor holding the relevant map row (Figure 3.5). Upon receipt of the predation message, P_j uses the panther's current coordinates to calculate where the search center should be on that processor (see Figure 3.6). The black grid cell in Figure 3.6 is the original location of the panther. The eight shaded grid cells surrounding that original location are the cells examined in the first search. The darker shaded grid cell is then chosen as the new search center. A new search is then conducted on the eight grid cells surrounding the new search center. One row of grid cells is on P_j, so a search message is sent to P_j so that a search may be conducted there. The darker shaded grid cell on P_j represents the grid cell that is calculated to be the new search center. The shaded grid cells on P_j represent the grid cells that will be examined in the first search on P_j. Processor P_j initiates a search from this new search center, and examines the surrounding grid cells as explained above (Figure 3.4). If the search pattern on P_j requires information about a grid cell not located on P_j, a message will be sent to the processor containing the relevant grid cell. The results of P_j's expanded search are returned to P_i, the processor that initiated the original expanded search, so that P_i may compare all search results before selecting a new

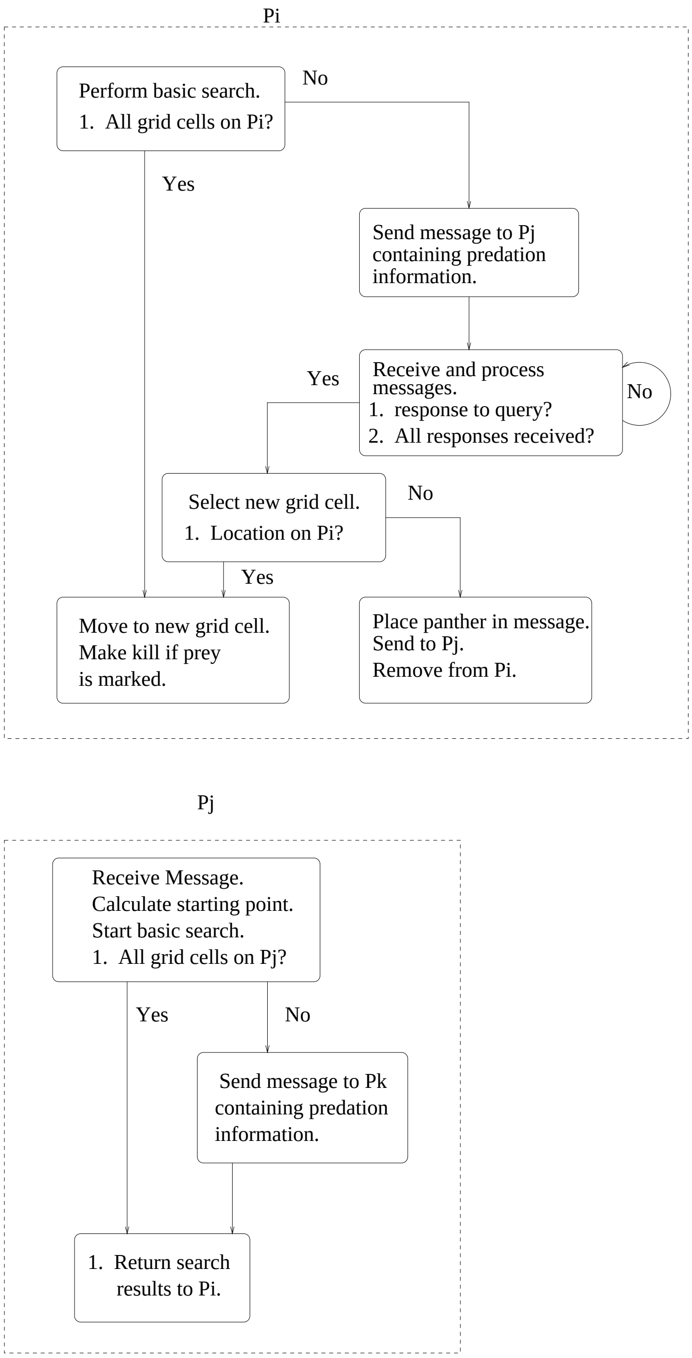

Figure 3.5: Expanded Predation Flow

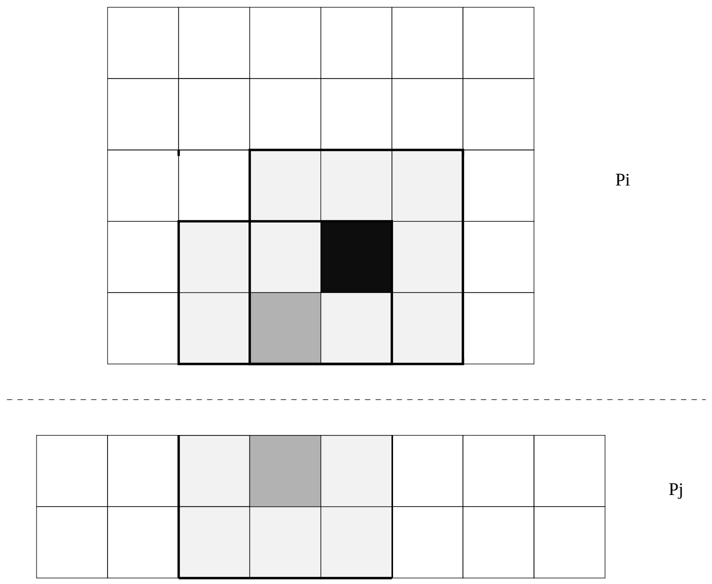

Figure 3.6: Predation Search Pattern

grid cell location for the panther that is searching. When the search is concluded, P_j then returns the information from the search to P_i so that the new grid cell may be selected and any prey may be removed.

If any predation messages were sent during the predation process, then P_i must wait for the results of those searches before making a kill or choosing a new location for the panther. As each message is returned, the maximum distance traveled and the type of prey are compared. The highest priority is a grid cell where potential prey are located. If more than one move allows for predation, the grid cell closest to the current location is selected as the new grid cell for the panther. When two potential locations are the same distance from the current grid cell, priority is given to the pixel with the highest concentration of available deer or hogs, if no deer were present. Selecting a new grid cell for the panther on P_i means that the row and column values of the panther are updated and any prey that were selected during the search are removed. If a deer, panther, or hog was not selected for predation and the panther is alive and not displaced by another panther, then the number of hogs in the new grid cell is evaluated and one is selected as prey.

Any predation or dispersal actions are delayed until after the final selection of a new grid cell location for the panther is made. If such actions were taken immediately without waiting for pending messages, the wrong prey might be removed or the panther might be relocated unnecessarily. If a final grid cell on another processor is chosen as a new location for the panther, only pending actions on the new processor will be performed. Selecting a location on another processor requires sending a panther message with the panther and dependent young to the new location. The predation process for the panther begins again on this new processor, with the new location serving as the initial search center. The predation search must be performed again because the intended prey may be removed by another panther while the predation search and relocation process for this panther was completed.

3.4.3 Dispersal

Panther in the DSIMPDEL model disperse under the same conditions as the serial SIMPDEL model. Dispersal occurs when panther leave the natal range at the age of 18 months, when predation fails for short or long range dispersal periods, or upon encountering another panther during predation. Two types of dispersal are used in the panther component: short range and long range. A short range dispersal is used when leaving the natal range and when another panther is located in a grid cell that is marked as a search center during the predation process. A long range dispersal occurs only when predation fails for 21 days or no acceptable pixels are located in the short range dispersal. Short and long range dispersals use the same searching process, they are differentiated only by the maximum distance to the new grid cell [CGF+96].

Dispersal begins by randomly selecting a grid cell within the panther's dispersal distance. In Figure 3.7, each square represents one $500m$ grid cell. The black square represents the grid cell that is the panther's initial location, the shaded squares are randomly generated locations, with the final destination highlighted in a lighter shade. New locations on P_i require only a check of the vegetation and water level of the grid cell to determine if the values are appropriate given the panther's gender and age. Grid cells not located on P_i require the use of a disperse message. The panther and any dependent young are sent in a message

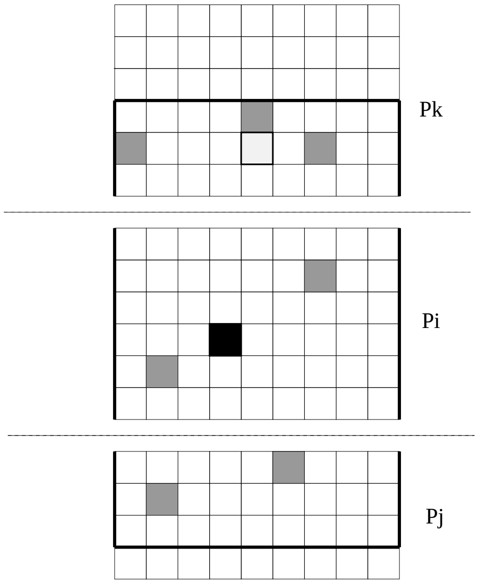

Figure 3.7: Dispersal Search Pattern

to P_j, the processor containing the grid cell being evaluated. Processor P_j first determines if the selected cell contains both an appropriate water level and suitable vegetation and then returns a message to inform P_i of the search results. If the new location is acceptable, P_i then removes the panther from the old location, the search is terminated, and the panther is added into the population of P_j. When a search is unsuccessful, P_j removes the panther and returns a message to P_i. If the maximum number of dispersal attempts is not exceeded, P_i generates another dispersal location and the dispersal process continues.

3.5 Synchronization

Both the deer and panther components allow each processor to work independently with synchronizations at points during the day to simplify message passing. In both PSIMDPEL and DSIMPDEL the context in which a message is sent requires no clarification, as all processors are at the same section of the model. An important reason for these synchronizations is to insure the consistency of the model. All messages could be sent and processed at one point each day, allowing less synchronization and communication. However, this would lead to inaccurate results because the decisions made for each animal impact the others in the population. For example, deer and panther are dispersed at the start of each day upon reaching the age of independence. Those animals are used throughout the simulation, completing each of the daily activities. If those animals were not processed until a later time, the state of the simulation would be changed. Those animals would not be available for reproduction and predation, and the deer would not forage. To achieve results similar to the serial SIMPDEL model, the decisions for each animal dispersed would be made at one time, and then the impact of those decisions would be propagated throughout the landscape, resulting in the re-processing of many animals. This would be highly inefficient and could result in inaccurate outputs.

In PSIMPDEL these synchronizations are performed with four CMMD functions: CMMD_SYNCH_WITH_NODES_START, CMMD_SYNCH_WITH_NODES_FINISH, CMMD_SET_GLOBAL_OR, and CMMD_GET_GLOBAL_OR [Abb95]. The functions CMMD_SET_GLOBAL_OR and CMMD_GET_GLOBAL_OR are used on the Thinking Machines CM-5 to allow processors to continue processing messages in a loop until all processors had completed the foraging phase [Abb95]. The value of CMMD_GET_GLOBAL_OR is a zero initially, as each processor completes the program section delineated by the synchronizing pair, it adds a one with a call to CMMD_SET_GLOBAL_OR. If the status of a processor changes, another call is made to CMMD_SET_GLOBAL_OR, and a zero is added to value of the global or. This continues until all processors complete the program section and the value returned by CMMD_GET_GLOBAL_OR is a one [Thi93]. The CMMD functions CMMD_SYNCH_WITH_NODES_START and CMMD_SYNCH_WITH_NODES_STOP are also used in tandem to ensure that the processors are synchronized while completing a section of the deer component. These two functions are used in the main function of the deer code to provide synchronization during the dispersal and reproduction sections of the deer component [Abb95].

The synchronizing functions described above, or any with similar functionality, are not available in MPI. A blocking barrier, such as the MPI_BARRIER function is not appropriate, because upon reaching such a barrier, the processor cannot continue until all the processors make the barrier call. This prevents a processor from receiving and responding to messages once it completes processing the ani-

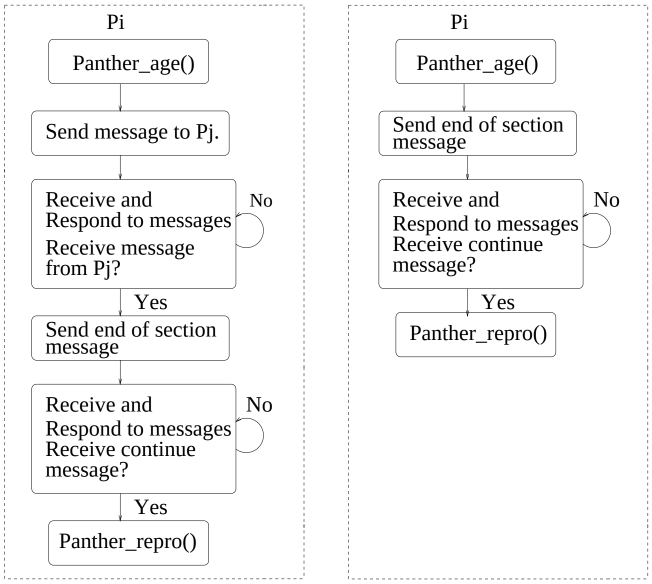

Figure 3.8: Synchronization process

mals in its population. Instead the synchronizations were implemented with MPI messages (Figure 3.8). Each process records the number of messages sent and received each day. When a process completes a section of the simulation, it sends a section completion message to P_0 that includes the number of sends and receives handled. Each process continues to poll for messages and updates P_0 when it sends or receives a new message. When all processes have sent a synchronization message indicating completion of the section to P_0, and the total number of receives is equal to the the total number of sends, P_0 sends a message to indicate that all processors may move to the next section. This method provides the same synchronizations as performed in the PSIMPDEL model with CMMD commands.

However, the CMMD synchronizations are performed in hardware, [Thi93] and are thus much more efficient than the MPI method which may suffer from high idle times.

Chapter 4

Verification and Performance Results

The outputs of the vegetation, hydrology, and deer components of the DSIMPDEL model were verified by comparing three selected outputs with values produced by runs of the serial SIMPDEL model. The performance of the DSIMPDEL implementation without the panther component was evaluated and compared with the performance of both the PSIMPDEL and serial SIMPDEL implementations. The performance results of the DSIMPDEL model with the panther component are also given. These verification and performance results are presented in the following sections.

4.1 Verification of Deer Component

The method of verification used for DSIMPDEL model is the same as the method employed to evaluate the outputs of PSIMPDEL [Abb95]. Three main statistics produced by the evaluation of outputs from runs of DSIMPDEL were compared with the results produced by SIMPDEL. These three selected outputs are average daily travel distance of deer, deaths due to weight loss, and year-end population size [Abb95]. These three statistics allow the verification of the hydrology, vegetation, and deer components of the model. The hydrology and vegetation components are verified through the average daily travel distance and the weight loss statistics. The average daily travel distance and the number of weight loss deaths are dependent upon the amount of forage available to the deer population. Since the amount of available forage is determined by the outputs from the vegetation and hydrology components, inaccuracy in either component would produce an unexpected variation in the weight loss death or average travel statistics.

The deer component is verified through the year-end population statistics. The population numbers directly validate the reproduction component, but also indirectly validate the dispersal, aging, and mortality components. The reproduction component directly influences the population size by determining the number of births per year. The aging component determines when animals are available for reproduction. Inaccuracies in dispersal could result in animals being relocated incorrectly which would affect the process of searching for a mate. Inaccuracies in the mortality component could produce inappropriately inflated or reduced population values.

The statistics shown in Figures 4.1 through 4.3 were calculated from the outputs of experiments using 12 processors (workstations) with an initial population of 10,000 deer. The maximum number of processors used for all experiments was 12. The statistics for the average daily travel distance of the deer obtained from experiments with the SIMPDEL and DSIMPDEL models are shown in Figure 4.1. The difference in the outputs of the SIMPDEL and DSIMPDEL models ranges from 0% in year two to a maximum of 8% in year 11. The log of the weight loss deaths per year for SIMPDEL and DSIMPDEL are shown in Figure 4.2. The `y-axis` of this graph is represented in log values because of the large range in the number of weight loss deaths per year. A total difference of 3% in the number of weight loss deaths during the entire 23 year simulation between the two models was observed. The year end population values for DSIMPDEL and SIMPDEL, which differ a by maximum of 4%, are shown in Figure 4.3. The results for experiments with initial populations of 2,000 and 20,000 deer are presented in Appendices A and B, respectively.

The deer distributions maps for selected years for the SIMPDEL and DSIMPDEL models are shown on pages 47-51. The selected maps represent the years with the greatest deer distribution changes from the previous year. Each colored

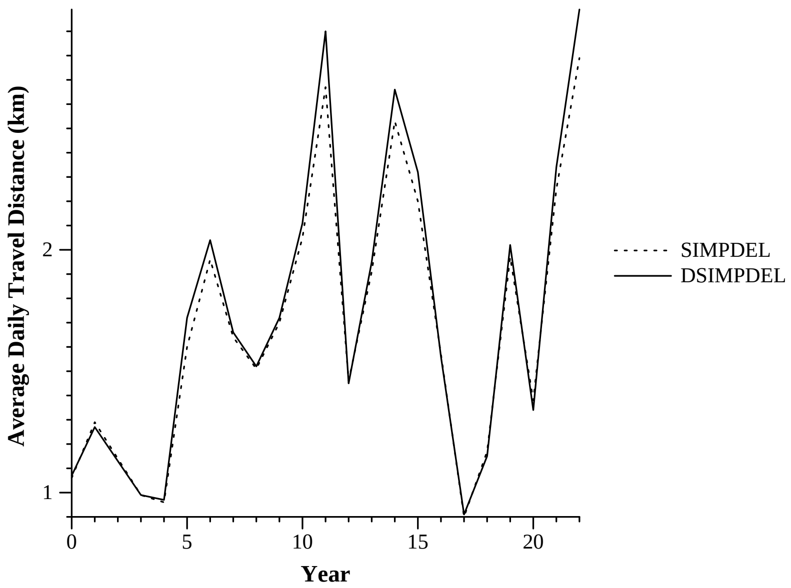

Figure 4.1: Average daily travel by year for an initial population of 10,000 deer.

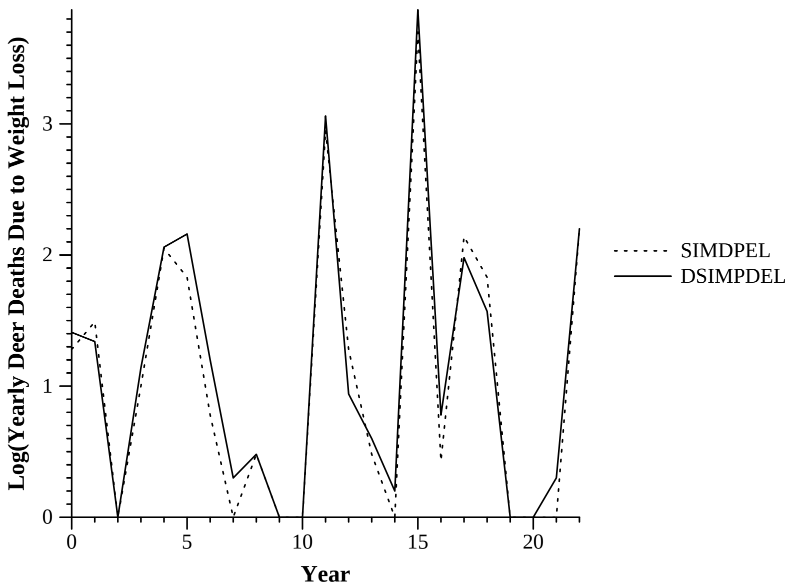

Figure 4.2: Weight loss deaths by year for an initial population of 10,000 deer.

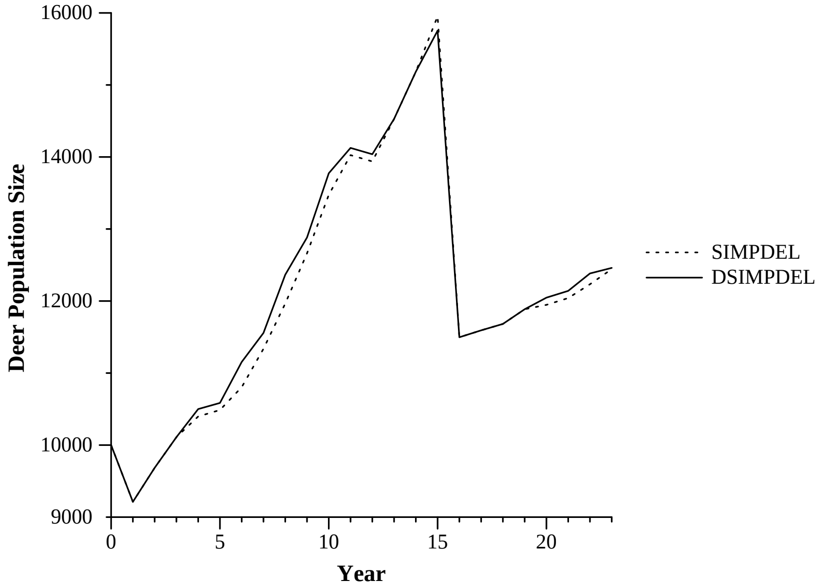

Figure 4.3: Year-end population values for an initial population of 10,000 deer.

dot represents one of the $500m$ grid cells. The number of deer located in the grid cell is represented by the color of the dot. A legend which lists the colors used in these maps is provided in Table 4.1. The initial random distribution of the deer population is shown in Figure 4.4. In both models, the initial distribution spreads the deer population evenly throughout the map. As illustrated in Figure 4.5, the deer distributions of both models (SIMPDEL and DSIMPDEL) at the end of year 4 are almost identical. The comparisons at the beginning of year 15 (Figure 4.6) and year 18 (Figure 4.7) show some differences in the distributions of deer, but the same general patterns are evident. By year 20 (Figure 4.8), the deer are once again distributed in a very similar manner.

Figure 4.4: Initial distributions for a population of 10,000 deer.

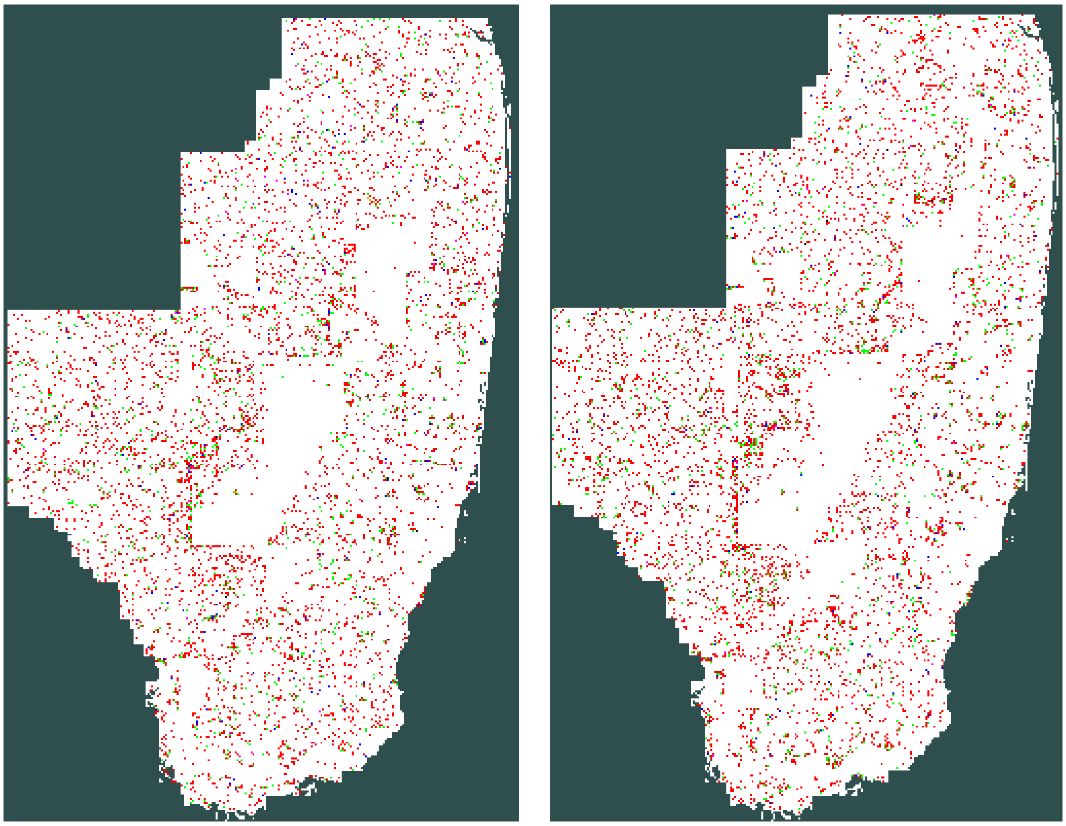

Figure 4.5: Deer distributions at the beginning of Year 4 for the serial SIMPDEL model and the DSIMDPEL model.

Figure 4.6: Deer distributions at the beginning of Year 15 for the serial SIMPDEL model and the DSIMPDEL model.

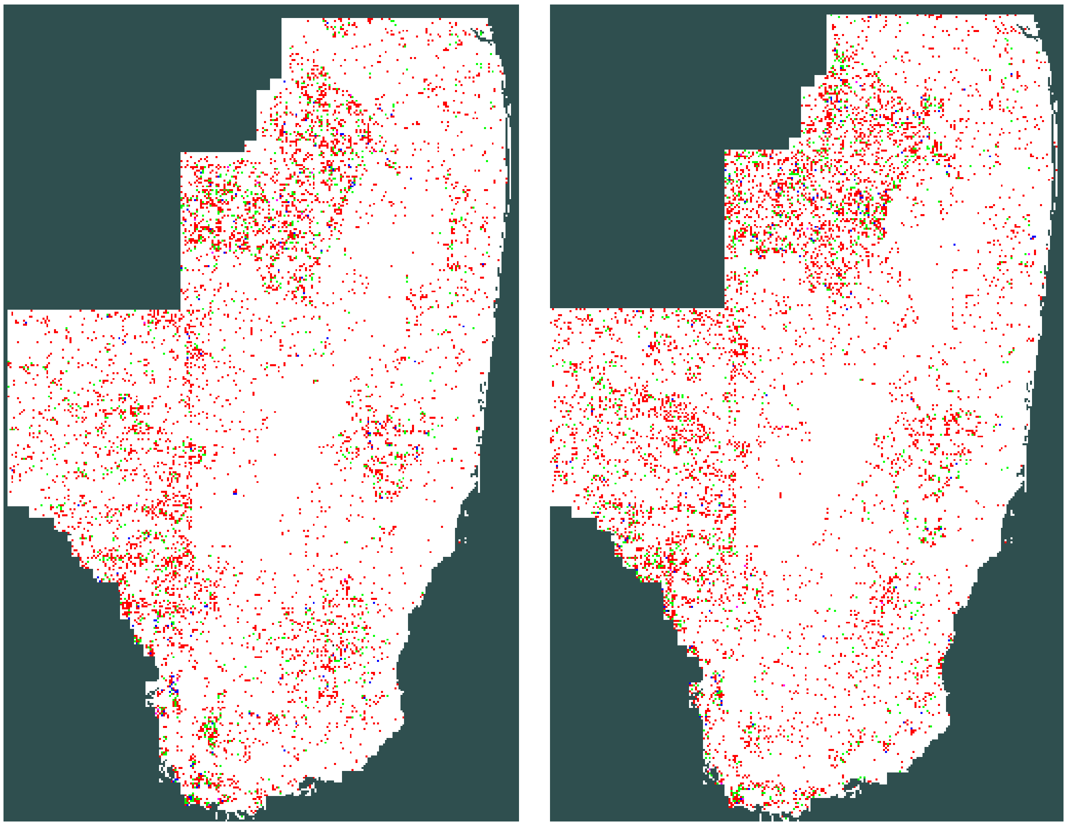

Figure 4.7: Deer distributions at the beginning of Year 18 for the serial SIMPDEL model and the DSIMPDEL model.

Figure 4.8: Deer distributions at the beginning of Year 20 for the serial SIMPDEL model and the DSIMPDEL model.

Table 4.1: Deer Distribution Map Legend

Color	Number of Deer
Red	1-2
Green	3-4
Purple	5-6
Pink	7-8
Black	9-10
Orange	11-20
Brown	21-30
Yellow	31-40
Blue	41-50

4.2 Performance Results

4.2.1 Computing Environment

The performance and verification results presented in this thesis were obtained on a network of Sun Microsystems Ultra 2 workstations, each containing two 167-Mhz UltraSPARC-1 processors under the Solaris 2.5.1 operating system. Each machine has 256-Mbytes of memory and two 2.1-Gbyte internal disks. Peak performance of one UltraSPARC-1 processor is 126 Mflops. The machines are connected by both a 10 Mbps Ethernet interface and 155 Mbps ATM sbus adapter. The executables and the input files were stored on the internal disk to eliminate additional network overhead. The serial SIMPDEL implementation tests were performed using one of the Sun Ultra 2 machines. The DSIMPDEL implementation was tested on configurations of four, eight, and twelve machines using both the Ethernet and the ATM connections. Only one of the two processors of the Sun Ultra 2 workstation was utilized during these experiments. The performance results for the PSIMPDEL implementation were obtained from experiments on a Thinking Machines CM-5 with 32 SPARC 2 processors [Abb95] [ABC+97]. Each of the 32 processors has 32 Mbytes of memory and a 32MHz clock. Maximum aggregate performance on this CM-5 is 4 Gflops.

4.2.2 Performance Methodology

The results presented below were obtained from experiments using three different initial populations of deer: 2,000, 10,000, and 20,000. Three tests were performed with each processor group for each population size and network type. Each result reported is an average of three runs. The times reported for the SIMPDEL and DSIMPDEL models are given in CPU time while times for PSIMPDEL are given in elapsed wall-clock time [Abb95]. The speed improvements reported were calculated with the formula $S(n) = T(1)/T(n)$ [Hwa93], where *T(n)* is the elapsed time on n processors (workstations).

4.2.3 Performance Results without the Panther Component

The greatest improvement in performance was achieved through the use of Sun Ultra 2 workstations. The serial SIMPDEL code used in the performance analysis is the same implementation that was used to verify the PSIMPDEL model [Abb95]. In [Abb95], performance results for SIMPDEL were obtained on a 85-Mhz Sun

SPARCstation 5 with 32 Mbytes of memory. The speedups obtained simply by executing the SIMPDEL model on a 167-Mhz Sun Ultra 2 with 256 Mbytes of memory is shown in Table 4.2. The times shown for the Sun SPARCstation 5 are represented in elapsed wall-clock time and were obtained from results presented in [Abb95]. The greatest speed improvements were seen in the vegetation and hydrology components. A speedup of 10.2 was seen in the vegetation component while the speedup for the hydrology component was 8.52. The largest speedup in the deer component was 9.66, which was seen with a initial population of 2,000 deer. These speedups are due both to the improved processor speed and the increased memory capacity of the Sun Ultra 2 workstation.

The execution times of the SIMPDEL, PSIMPDEL, and DSIMPDEL implementations with an initial population of 2,000 deer are shown in Figure 4.9. For this population size the greatest speed improvement over the SIMPDEL implementation was 3.96. This speed improvement was obtained on 12 processors using the ATM switch (Table 4.3). Times obtained from testing the three implementations with an initial population of 10,000 deer are shown in Figure 4.10, and the corresponding speed improvements are provided in Table 4.4. The execution times and speed improvements for an initial population size of 20,000 deer are shown in Figure 4.11 and Table 4.5, respectively.

The partitioning of the vegetation and hydrology data across the processors does result in speed improvements ranging from 3.88 on the 4-processor imple-

Table 4.2: Execution Times for the SIMPDEL model on the Sun SPARCstation 5 and the Sun Ultra 2 for initial populations of 2,000, 10,0000, and 20,000 deer.

Initial population of 2,000 deer.			
Component	Sun SPARCstation 5	Sun Ultra 2	Speedup
Deer	16071	1610	9.98
Hydrology	21345	2504	8.52
Vegetation	20713	2028	10.2
Total	59371 (16.5 hrs)	6142 (1.7 hrs)	9.66

Initial population of 10,000 deer.			
Component	Sun SPARCstation 5	Sun Ultra 2	Speedup
Deer	55482	9224	6.0
Hydrology	21354	2504	8.52
Vegetation	20713	2028	10.2
Total	98057 (27.2 hrs)	13756 (3.82 hrs)	7.12

Initial population of 20,000 deer.			
Component	Sun SPARCstation 5	Sun Ultra 2	Speedup
Deer	93235	21372	4.36
Hydrology	21354	2504	8.52
Vegetation	20713	2028	10.2
Total	140434 (39.0 hrs)	25904 (7.19 hrs)	5.42

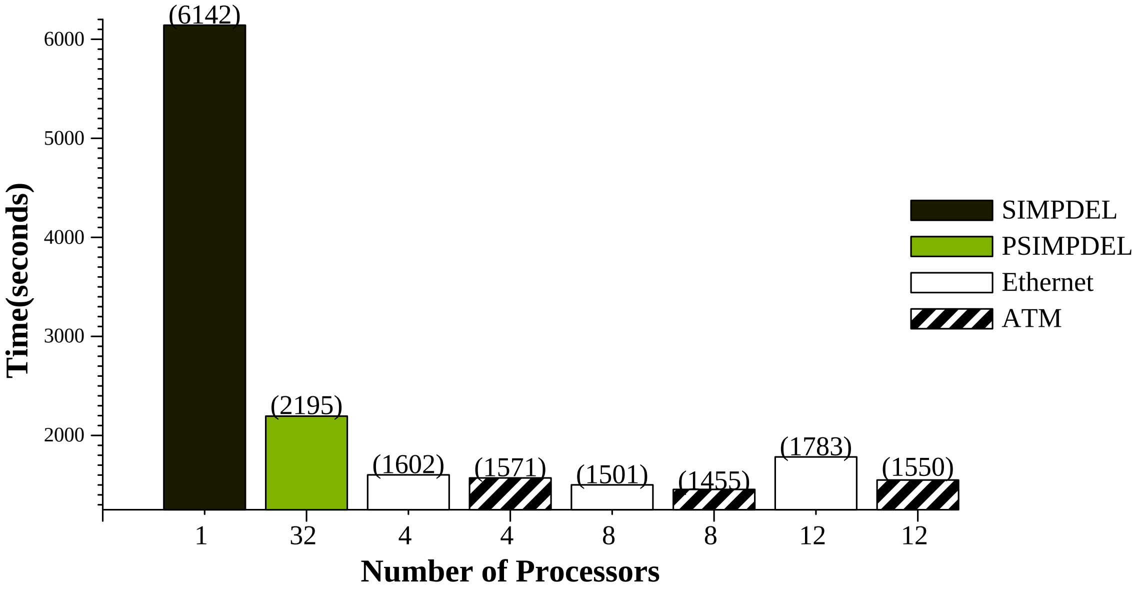

Figure 4.9: Comparison of execution times for runs with initial population of 2,000 deer.

Table 4.3: Average speed improvement for the PSIMPDEL and DSIMPDEL models with an initial population of 2,000 deer.

Model	Processors	Network	Speed Improvement
PSIMPDEL	32	N/A	2.79
DSIMPDEL	4	Ethernet	3.83
	4	ATM	3.90
	8	Ethernet	4.09
	8	ATM	4.22
	12	Ethernet	3.44
	12	ATM	3.96

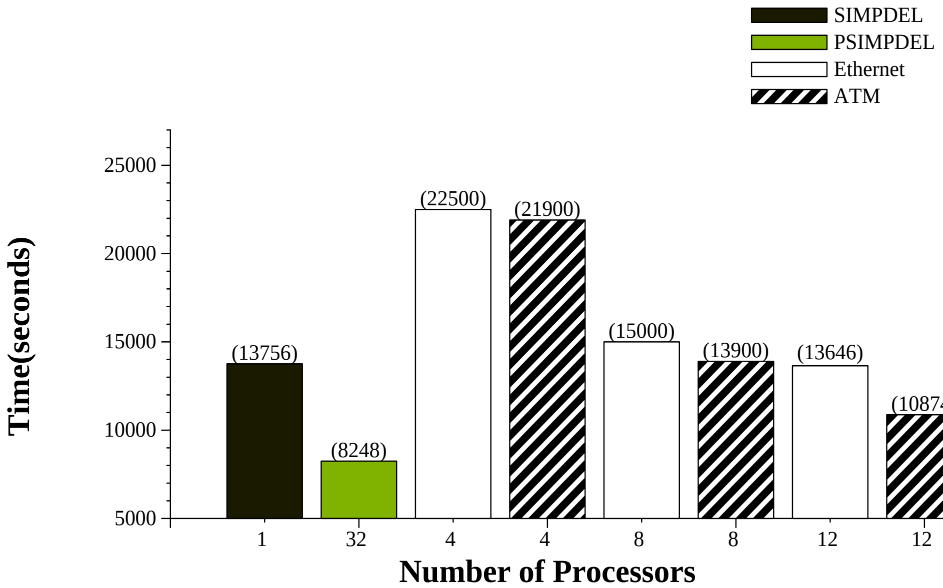

Figure 4.10: Comparison of execution times for runs with initial population of 10,000 deer.

Table 4.4: Average speed improvement for the PSIMPDEL and DSIMPDEL models with an initial population of 10,000 deer.

Model	Processors	Network	Speed Improvement
PSIMPDEL	32	N/A	1.67
DSIMPDEL	4	Ethernet	.611
	4	ATM	.628
	8	Ethernet	.870
	8	ATM	.911
	12	Ethernet	1.01
	12	ATM	1.27

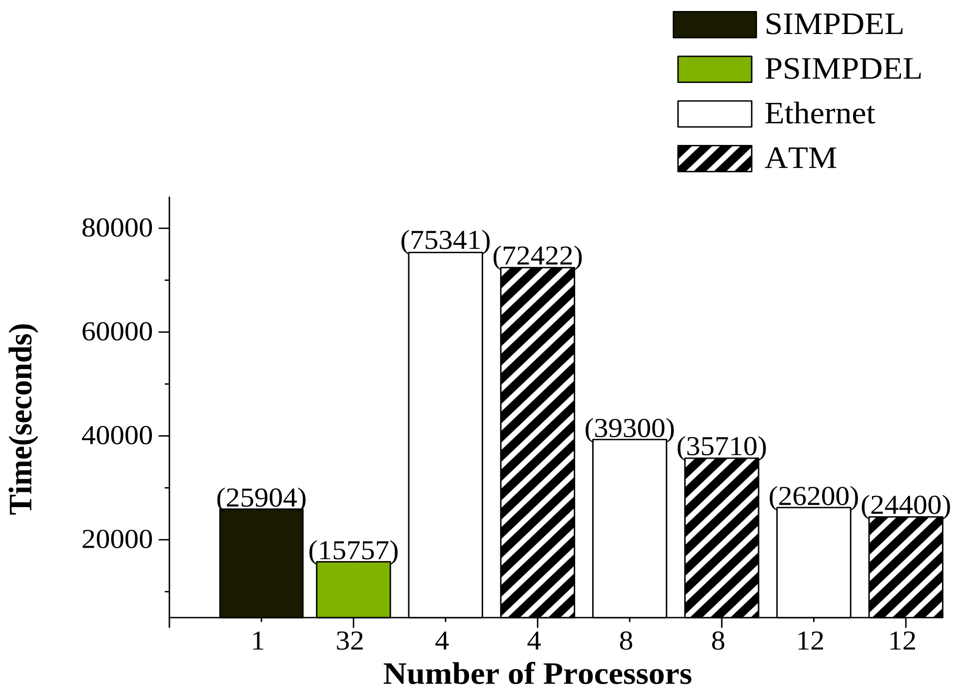

Figure 4.11: Comparison of execution times for runs with initial population of 20,000 deer.

Table 4.5: Average speed improvement for the PSIMPDEL and DSIMPDEL models with an initial population of 20,000 deer.

Model	Processors	Network	Speed Improvement
PSIMPDEL	32	N/A	1.64
DSIMPDEL	4	Ethernet	.344
	4	ATM	.357
	8	Ethernet	.659
	8	ATM	.725
	12	Ethernet	.988
	12	ATM	1.06

mentation to 14 with the 12-processor implementation. However, these speed improvements are offset by the communication costs incurred in the deer component. The maximum speed improvement in the deer component is 1.2 which was obtained with the 12-processor DSIMPDEL implementation. The communication costs are represented by the processor *idle time*, the amount of time a processor spends waiting for deer and synchronization messages during each simulation. The increase in processor idle times incurred by message passing and the synchronization process described in Section 3.5 degrade the performance results for all three initial population sizes. Processor synchronization after each phase of the deer component is necessary to ensure that all message-passing is completed at the conclusion of that phase by any processor. This constraint ensures both that all messages are sent and received in the same context and that no messages are sent to processors which have completed the current phase. Without the synchronization process, messages sent to processors which had completed the current phase would not be received. The simulation would fail as the sending processor waited for a message response. The relationship between average communication and computation costs for initial deer population sizes of 2,000 and 10,000 deer is illustrated in Figures 4.12 and 4.13. Computation and communication curves similar to Figure 4.13 were obtained with initial deer populations of 20,000 deer. For all three initial deer population sizes the computation time decreases as the number of processors increase. This decrease in computation time reflects the

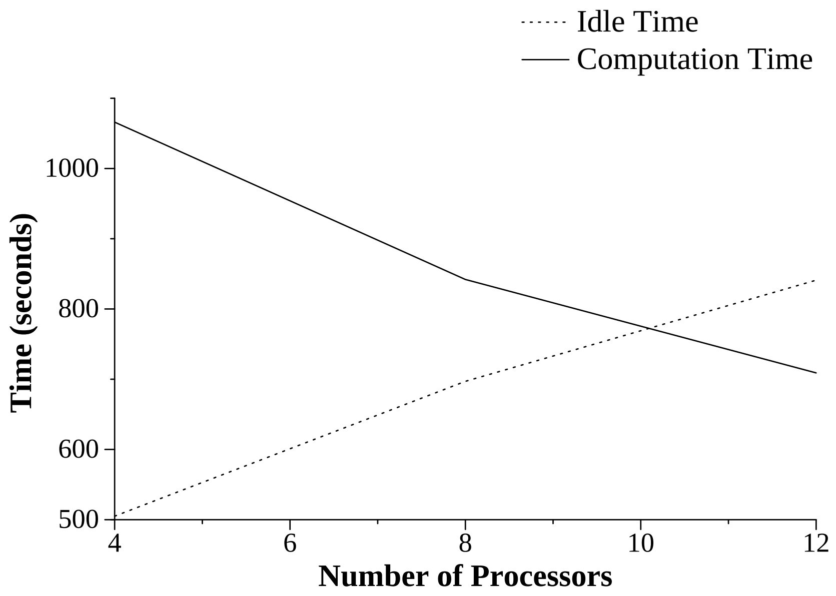

Figure 4.12: Comparison of computation time versus idle time for an initial population of 2,000 deer.

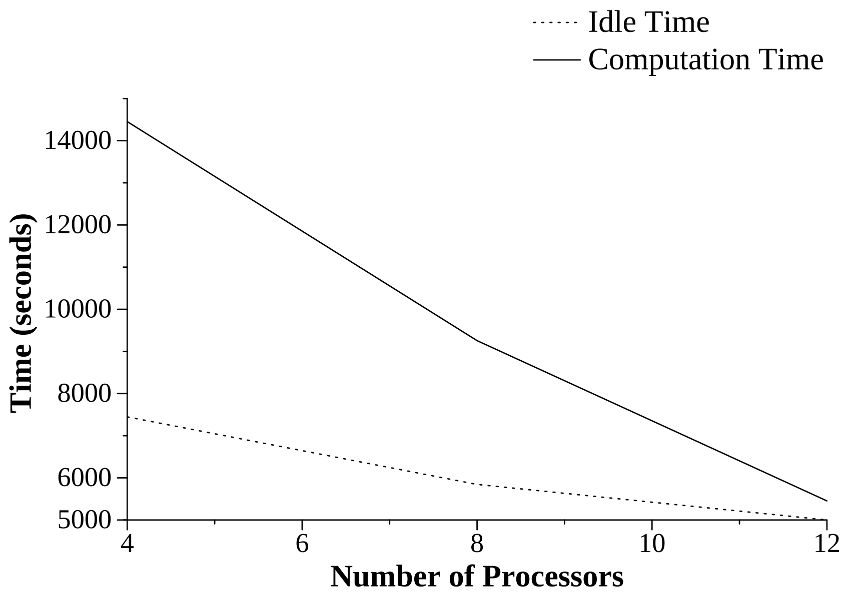

Figure 4.13: Comparison of computation time versus idle time for an initial population of 10,000 deer.

reduction in the number of rows owned by each processor. The decrease in the number of rows reduces the number of hydrology and vegetation calculations, and the size of the deer population on each processor. Table 4.12 shows the idle time increasing with the number of processors for an initial population of 2,000 deer. This is a result of an increase in the portion of time spent in the synchronization process. Table 4.13 shows a similar decrease in computation times as the number of processors increases with an initial population size of 10,000 deer. However, for this population size, the idle time decreases with larger numbers of processors. This decrease in idle time can be attributed to more favorable load balancing associated with the distribution of deer across the processors. The load-balancing inequities are most pronounced in the 4-processor configuration and are reduced as the number of processors increases. An example of the relationship between computation time and the number of simulated deer on 8 processors is shown in Figures 4.14 and 4.15.

4.2.4 Performance Results with the Panther Component

The results of testing the DSIMPDEL model with both the deer and panther components provided results similar to those obtained with only the deer component. The testing of the panther component was performed with the three initial deer population sizes used in the experiments discussed in Section 4.2.3 and an initial population of 30 panther. The execution times for the experiments with an

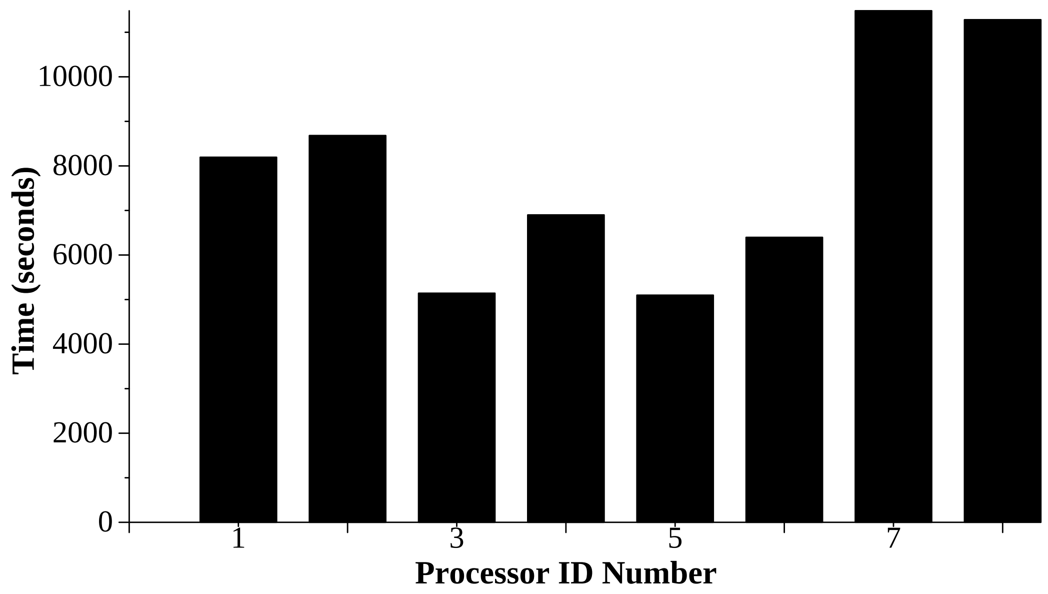

Figure 4.14: Total computation time per processor for an initial population of 10,000 deer.

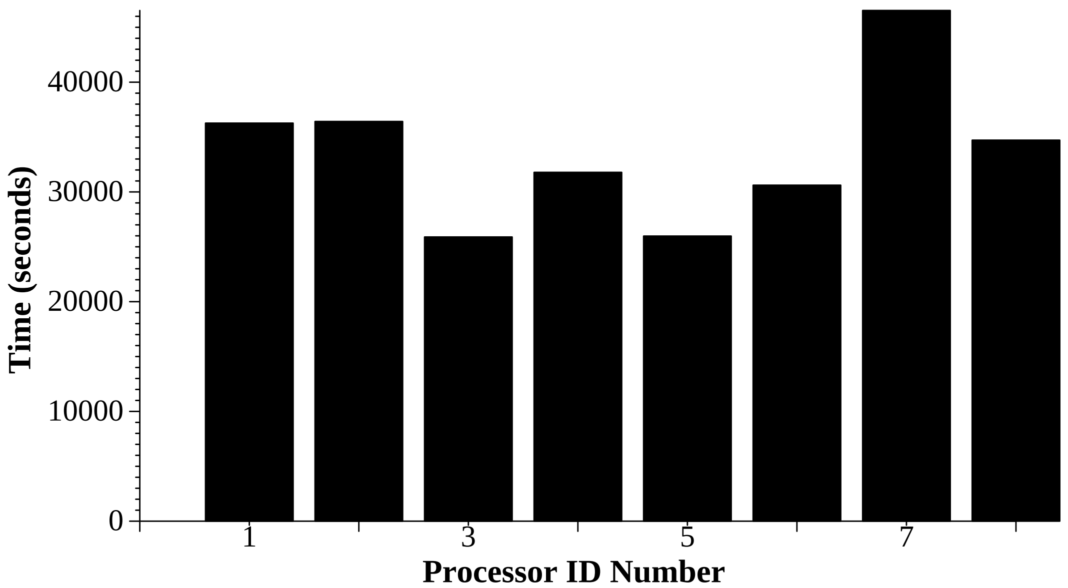

Figure 4.15: Total number of deer handled per processor for an initial population of 10,000 deer.

initial population of 2,000 deer are shown in Figure 4.16. The execution times for the deer component are lower than those observed in the experiments discussed in Section 4.2.3. This is due to the effect of the panther on the deer population. Specifically, the panther rely heavily upon the deer population as a source of prey, therefore the deer population values are smaller than those seen in deer-only testing. The execution times for the experiments with an initial population of 10,000 deer are illustrated in Figure 4.17. For larger deer populations, deer are located by panther more quickly during the predation process. This results in reduced execution times for the panther component. The execution times for the deer population, however, are slightly increased since the panther do not kill deer at a rate that maintains the population levels seen in the deer-only testing. Similar performance results for both deer and panther components were obtained for initial deer populations of 20,000.

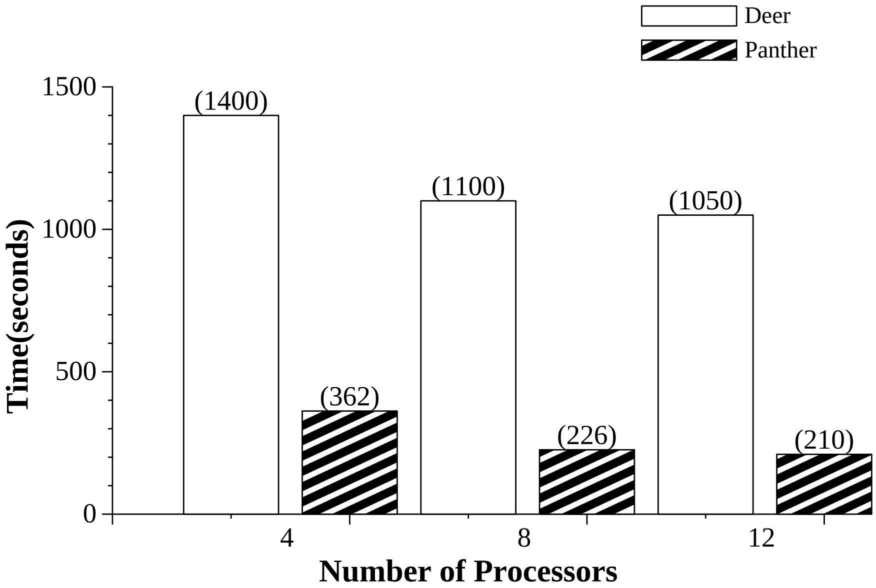

Figure 4.16: Comparison of execution times for runs with an initial population of 2,000 deer and 30 panther.

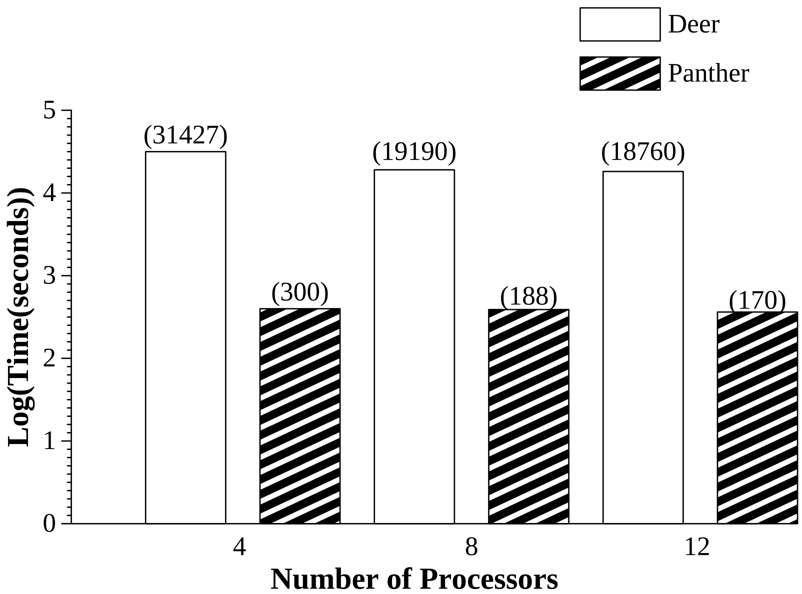

Figure 4.17: Comparison of execution times for runs with an initial population of 10,000 deer and 30 panther. Actual time in seconds shown in parentheses

Chapter 5

Summary and Future Work

This thesis has presented DSIMPDEL, the distributed implementation of the vegetation, hydrology, white-tailed deer, and Florida panther components of the SIMPDEL model. DSIMPDEL provides survival statistics and bioenergetic results consistent with the serial SIMPDEL model and moderate speed improvements ranging from 3.44 to 4.22 for small initial deer populations. The performance of the DSIMPDEL model was degraded by the high communication costs incurred by the synchronization process. Currently, the synchronization process prevents the simulation from failing when messages are sent to processors that have completed the current phase of the model and would therefore be unable to receive messages or return responses to the sender. The synchronization process could be made less stringent if a degree of fault-tolerance was incorporated in such cases to prevent the sender from waiting indefinitely for a message response. This high synchronization cost was most pronounced on large deer populations, for small deer populations the DSIMDPEL model provided the greatest speed improvements over the serial SIMPDEL model. The high cost of the synchronization process on larger deer populations suggests that the hardware-supplied synchronization implemented on shared-memory multiprocessor architecture is preferable to message-passing on a network of workstations. However, the price/availability of a shared-memory multiprocessor system does not necessarily make that approach cost-effective.

Future work to the model includes additions to the landscape representation to reflect the vegetation and hydrology changes caused by Hurricane Andrew. The panther component in both SIMPDEL and DSIMPDEL must be modified to simulate the territorial behavior of the panther population. This change will impact the panther movement patterns. Since changes in panther movement have the potential to affect the predation patterns, repartitioning the landscape may be needed to ensure equitable load-balancing across the processors. Repartitioning is not currently needed as predation does not drastically affect the equitable distribution of deer across the processors (assuming more than 4 processors are used). Dynamic load balancing in response to changes in deer distribution will certainly incur costs. The relocation of a map row requires the movement of data (approximately 500k bytes) associated with the deer and panther positioned in that row and the corresponding vegetation and hydrology data.

Bibliography

Bibliography

[Abb95] C. Abbott. A Parallel Individual-Based Model of White-Tailed Deer in the Florida Everglades. Master's thesis, The University of Knoxville, Tennessee, Knoxville, TN, 1995.

[ABC+97] C.A. Abbott, M.W. Berry, E.J. Comiskey, L.J. Gross, and H.-K. Luh. Computational Models of White-Tailed Deer in the Florida Everglades. *IEEE Computational Science and Engineering*, 1997. To be published.

[Bel89] R. Belden. The Florida Panther. In *Audubon Wildlife Report*, pages 525–532. National Audubon Society, New York, N.Y., 1986-1989.

[CGF+96] E. Comiskey, L. Gross, D. Fleming, M. Huston, O. Bass, H.-K. Luh, and Y. Wu. A Spatially-explicit Individual-based Simulation Model for Florida Panther and White-tailed Deer in the Everglades and Big Cypress Landscapes. In *Proceedings of the Florida Panther Conference, November 1994*. Ft. Myers FL, 1996. To be published.

[FDG+94] D. Fleming, D. DeAngelis, L. Gross, R. Ulanowicz, W. Wolff, W. Loftus, and M. Huston. *ATLSS: Across-Trophic-Level System Simulation for the Everglades and the Big Cypress Swamp.* National Biological Survey, ATLSS Workshop, Homestead, FL, June 1994.

[Hwa93] K. Hwang. *Advanced Computer Architecture.* McGraw-Hill, Inc., New York, NY, first edition, 1993.

[SB94] T. Smith and O.L. Bass. Landscape, White-Tailed Deer, and the Distribution of Florida Panthers in the Everglades. In *Everglades: The Ecosystem and Its Restoration*, pages 693–708. St. Lucie Press, Delray Beach, Florida, 1994.

[SOHL+96] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra. *MPI: The Complete Reference.* The MIT Press, Cambridge, MA, first edition, 1996.

[Thi93] Thinking Machines Corporation, Cambridge, Massachusetts. *CMMD Reference Manual Version 3.0*, May 1993.

Appendices

Appendix A

Verification Results from Testing with an Initial Population Size of 2,000 Deer

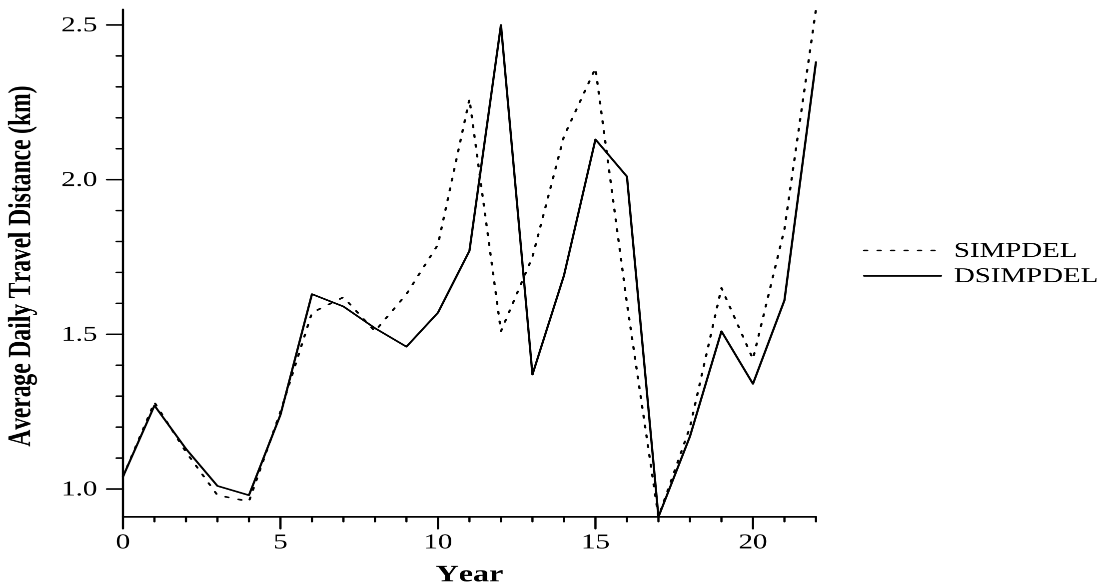

Figure A.1: Average daily travel by year for an initial population of 2,000 deer.

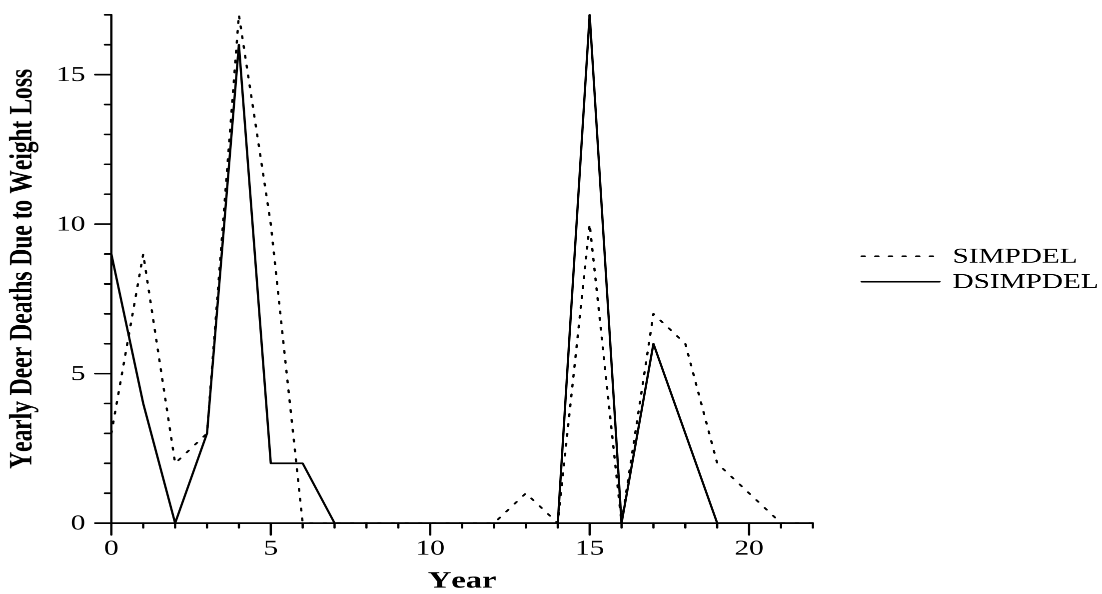

Figure A.2: Weight loss deaths by year for an initial population of 2,000 deer.

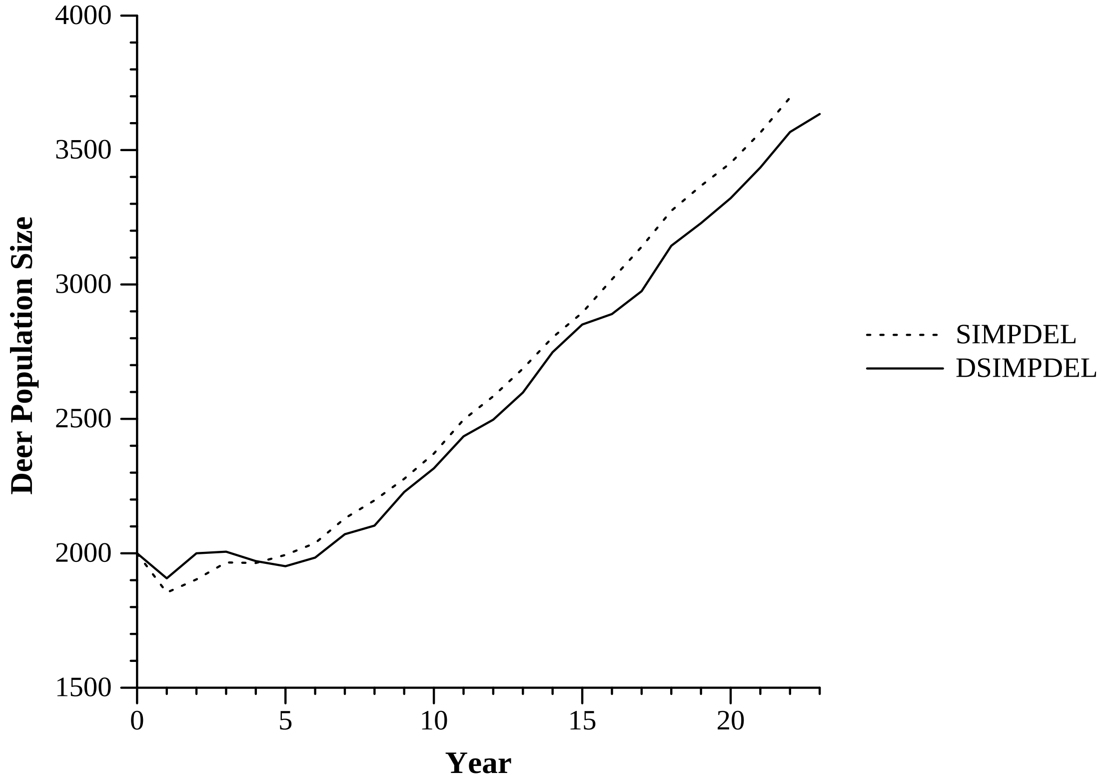

Figure A.3: Year-end population values for an initial population of 2,000 deer.

Appendix B

Verification Results from Testing with an Initial Population Size of 20,000 Deer

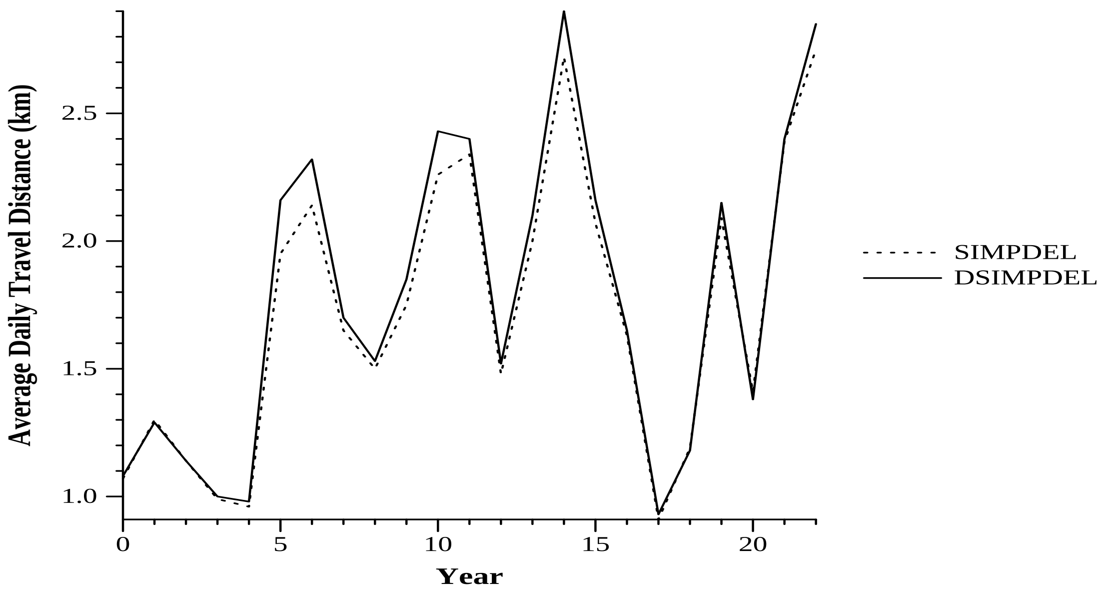

Figure B.1: Average daily travel by year for an initial population of 20,000 deer.

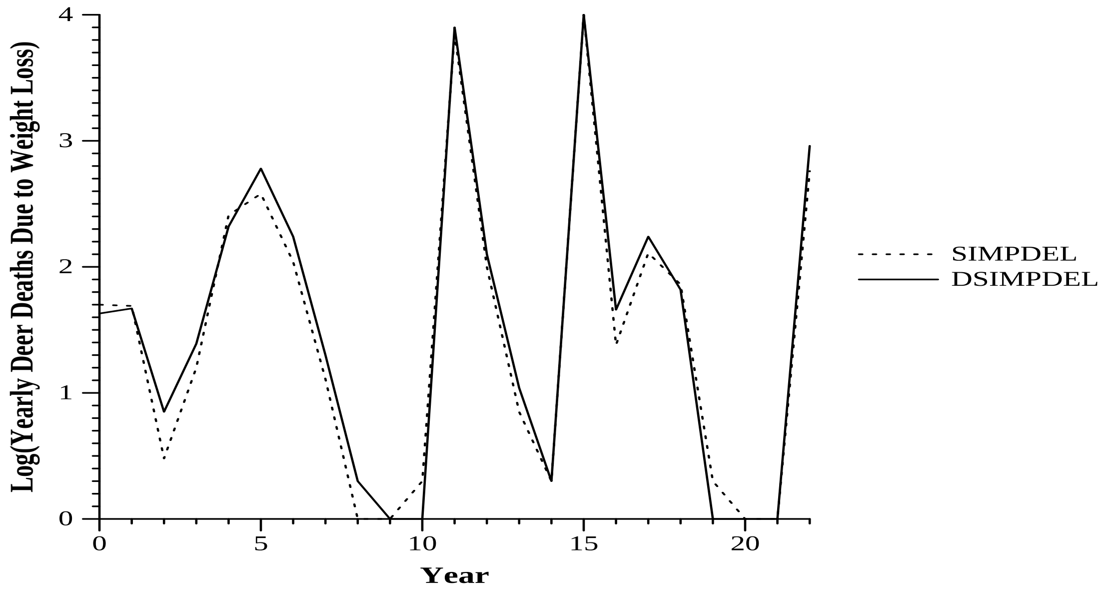

Figure B.2: Weight loss deaths by year for an initial population of 20,000 deer.

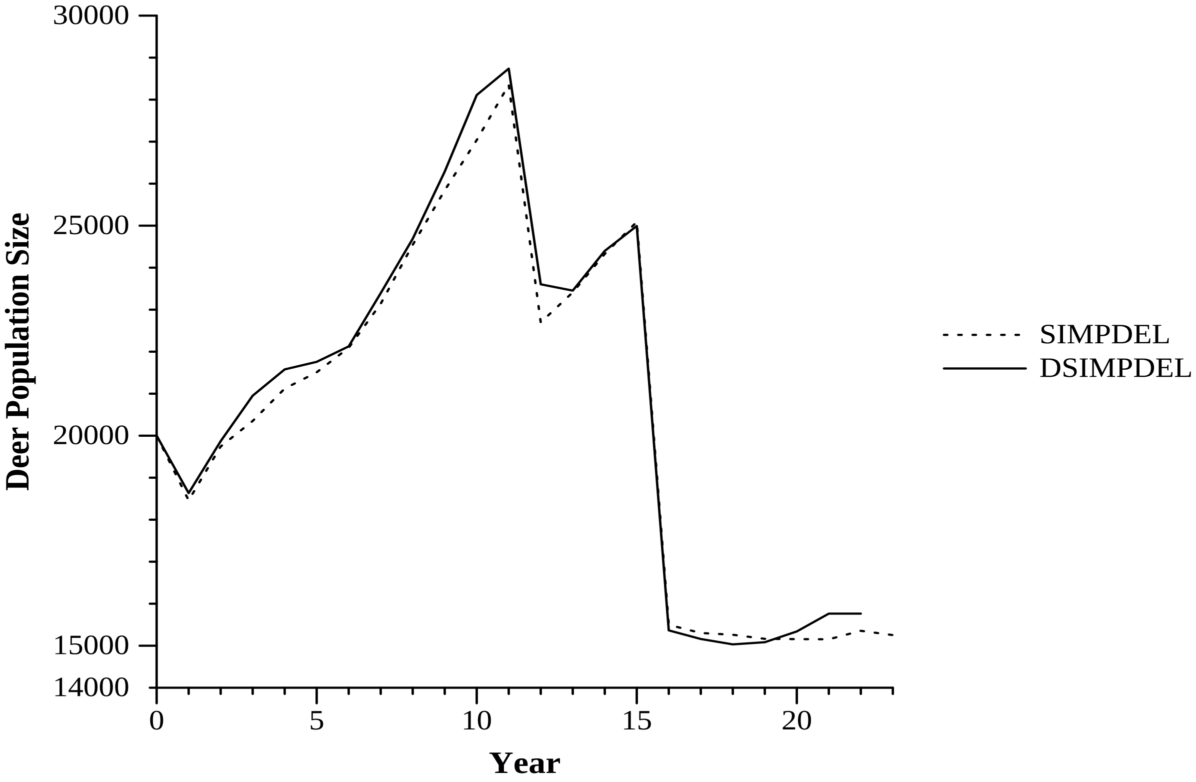

Figure B.3: Year-end population values for an initial population of 20,000 deer.

Vita

Linda Ellen Mellott was born in Everett, Pennsylvania on June 6, 1972. She graduated from Everett Area High School in 1990, and received a Bachelor of Science degree in Computer Science in 1994 from Dickinson College in Carlisle, Pennsylvania. In August 1994, she entered the graduate program in Computer Science at the University of Tennessee and was awarded the Master of Science degree in Computer Science in May 1997.