Parallel Models
of Animal Migration in
Northern Yellowstone National Park

Ember Uziel & Michael W. Berry

Computer Science Department

CS-94-262 November 1994

Submitted to The International Journal of Supercomputing Applications and High Performance Computing, (1995)

Parallel Models of Animal Migration in Northern Yellowstone National Park

EMBER UZIEL uziel@cs.utk.edu
Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville TN 37996-1301

MICHAEL W. BERRY berry@cs.utk.edu
Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville TN 37996-1301

Summary

In landscape ecology, computer models are developed to simulate the migration of animal groups as they exist in nature. Typically, these models are sequential, and the animal groups move and forage in sequence. In this work, we discuss a parallel implementation of the NOYELP (NOrthern YELlowstone national Park) model on a 32-processor Thinking Machines CM-5. NOYELP is a spatially-explicit individual-based model that simulates the search, movement and foraging activities of groups of animals across the northern portion of Yellowstone National Park. NOYELP is primarily used to study the effects of fire scale and pattern on the winter foraging dynamics and survival of free-ranging bison and elk groups. Separate versions of a parallel NOYELP model (referred to as PNOYELP) with different degrees of data-sharing and message-passing are used to study the effects on travel distances and semi-annual energy gain statistics. Comparisons of these statistics with the sequential NOYELP model are made, and reasonable speed improvements (ranging from 2.8 to 6.7) for PNOYELP on the CM-5 over the sequential model on a Sun SPARCstation 10 are reported.

Introduction

Computer models can be used to study the responses of ungulates to environmental heterogeneity (Turner et al., 1994). Models from landscape ecology, in particular, attempt to simulate interactions between animals and vegetation. The NOYELP (northern Yellowstone National Park) landscape ecology model used was originally developed by Drs. Monica Turner and Yegang Wu of the Environmental Sciences Division, Oak Ridge National Laboratory. NOYELP is a spatially-explicit individual-based simulation model developed to explore the effects of fire scale pattern on the winter foraging dynamics and survival of free-ranging elk (*Cervus elaphus*) and bison (*Bison bison*) in northern Yellowstone National Park. The model is used to project ungulate survival on the winter range under different fire regimes and winter weather scenarios (Turner et al., 1994).

The primary goal of this research was to reduce the effective run time of the NOYELP simulation via parallel processing, while maintaining the same trends in selected semi-annual (i.e., for 180 days) energy balance statistics observed with the sequential model. Because of the stochastic nature of NOYELP, it required multiple simulations that often demanded several hours of run time on RISC-based workstations. Until now, no other related landscape ecology models have been implemented on a multiprocessor architecture; thus, another motivation for developing the parallel NOYELP model (or PNOYELP) was to examine the effect of allowing the animal groups to move in parallel as opposed to the sequence dictated by the sequential model. In designing PNOYELP, assumptions (i.e., determining animal movement, animal moving order, and animal search area) were made that were not in the sequential model. Before discussing the sequential model (NOYELP) in Section 1, the remainder of this section introduces the MIMD architecture used to develop the PNOYELP model: a 32-processor Thinking Machines CM-5.

The 32 processing nodes (PNs) of the CM-5, which are scheduled by a single control processor (CP) (also known as the partition manager or PM), collectively form 1 partition of a *fat-tree* data network (Hwang, 1993). In general, the size of the partition can range from tens to thousands of processors. Each partition runs the CMOST operating system, an enhanced version of UNIX. With the vector unit accelerators (VUs) installed, each PN has four VUs that provide enhanced performance for parallel programs that are arithmetic in nature. Using the VUs, each PN is capable of achieving 128 megaflop/s (millions of floating-point operations per second) so that the peak performance of the 32-processor CM-5 is 4 gigaflop/s (billions of floating-point operations per second). In this work, however, the vector units were not used since a majority of the required computation does not require floating-point operations.

Access to the PNs in a partition is accomplished through the CP. Programming on the CM-5 utilizes one of two paradigms: *host/node* and *hostless*. In the *host/node* paradigm, one processor, usually the CP, acts as the *host* for the program and controls both the program flow and the workload assigned to each PN. Also in the *host/node* paradigm, the program is not limited to executing the same program on each of the PNs. It is feasible to execute different programs working on independent data on each of the PNs in the partition (MIMD). In the *hostless* paradigm, all of the PNs are executing the same program, on the same data or different data, in parallel without the influence of a CP. The CMOST operating system provides a limited *host* program to control message passing and I/O service request generated by the PNs. The *host/node* paradigm was selected in order to allow the animals to move independently on each PN.

Communication between the CP and the PNs or between individual PNs is accomplished through message-passing routines that are supplied by CMMD version 3.2. The Fortran-77 compiler (`f77`), supplemented with synchronous message-passing library routines, was used on each CM-5 node. Both point-to-point and broadcast message-passing paradigms were used in order to simulate animal movement and track daily energy balance statistics (discussed later in Section 2.4).

Results and Discussion

1. Northern Yellowstone National Park Model

(NOYELP) is a spatially explicit stochastic simulation model developed to explore the effects of fire scale and pattern on the winter foraging dynamics and survival of free-ranging elk and bison. The search, movement, and foraging activities of individuals or small groups of ungulates are simulated in the model. This section summarizes the NOYELP model originally discussed in Turner et al. (1994).

1.1. Study Area

The Yellowstone National Park (YNP) was established in 1872 and encompasses 9000 km^2 in the northwest corner of Wyoming and adjacent parts of Montana and Idaho. The NOYELP model focuses on the northern 20% of the Park, which covers 77,020 hectares. The climate is characterized by long, cold winters and short, cool summers (Diaz, 1979), (Dirks, 1982). About 83% of the elk's winter range, which is warmer and drier than the rest of the Park, is included within YNP. The vegetation consists of continuous forests at higher elevations and sagebrush-grasslands interspersed with coniferous forests at lower elevations. During 1988, approximately 34% of the winter range burned (Despain et al., 1989). Simulations used in this work reflect fire pattern data attained from the 1988 fires.

1.2. Model Description

The YNP landscape is influenced by different fire disturbances and winter severities which affect the foraging activities of bison and elk. Forage intake is a function of a group's initial body weight, the quantity of forage available on a site, and the depth and density of snow. The group's endogenous reserves are reduced to offset deficits when the energy expenditures of a group exceed the energy gained during a day.

The landscape is represented as a gridded, irregular polygon with a spatial resolution of 1 hectare. A time step of 1 day is used in the simulation for a total of 180 days spanning November 1 through the end of April. Because the model does not project ungulate reproduction or plant regrowth during spring and summer, simulations are conducted anew for each winter season.

1.2.1. Landscape Representation

Because of the irregular shape of the northern range, a 285 row by 584 column pixel map (or grid cell map) encompasses the 77,020 grid cells (Figure 1). Spatial heterogeneity across the landscape is represented by several data layers (i.e., fire pattern, slope, aspect, habitat, and elevation) obtained from the YNP geographic information system (GIS).

Figure 1. NOYELP landscape with 285 row × 584 column bounding box.

1.2.2. Forage Distribution

An initial quantity of available forage is assigned to each grid cell based on its habitat type and burn status at the beginning of each simulation (Minser, 1994). As the ungulate groups graze the forage on a given pixel, the absolute abundance of forage on that pixel is decremented. Since the model does not simulate regrowth or reproduction, forage abundance either remains constant or declines over the 180 day simulation.

1.2.3. Snow Simulation

Snow conditions are an extremely important determinant of winter ungulate dynamics, thus, the model must simulate the spatial and temporal heterogeneity of these conditions across the northern range. The northern range is subdivided into two subregions for the snow simulation. One subregion is in a rain shadow[1], thus it has low precipitation levels, while the rest of the landscape comprises the second snow subregion.

Snow simulation is a two-step process. In the first step, baseline snow depth and snow density, which are used to determine energetic cost of travel and the maximum daily moving distance, are projected within each of the two subregions. Monthly data for snow depth, snow-water equivalent, and weather are used for the baseline snow projections. Snow-water equivalent, an integrated measure of snow mass, is used to influence daily forage intake of the animals. The change in snow conditions from month to month is assumed to be linear. Although a 3-day interval for updating snow conditions is used in the simulations, the model allows the user to specify any time interval. The second step involves the distribution of snow to each grid cell. This process is accomplished by modifying the baseline projection based on the slope and aspect of each grid cell.

1.2.4. Ungulate Initial Conditions

Six ungulate classes are represented in the model: cows, calves, and bulls for both elk and bison. The herds of elk and bison are initialized with a total number of ungulates. Each herd is then divided into calf, cow, and bull groups that will remain together for the entire simulation. All ungulate groups are assigned an initial bodyweight based on their ungulate class. Since the animal groups move one by one, they are assigned a randomly generated number (*animal group number*) to determine their moving order. Each calf group is assigned to a cow group and moves accordingly to the same site.

Equations for updating snow and search, movement and foraging activities (see Uziel, 1994) apply to individual animals; the group size is used as a multiplier. Elk cows, calves, and bulls maintain a group size of 4 animals. Bison cows and calves have a group size of 9 and bison bulls travel in a group size of 2.

Initially, elk are placed randomly across the entire landscape on grid cells containing forage. Bison are also distributed randomly on resource sites, but only in nonforest habitats at elevations less than 2100 meters. More than one group of elk or bison may simultaneously occupy a single grid cell throughout the simulation.

1.2.5. Ungulate Foraging

If there is forage above a threshold (explained in Section 2) available on a grid cell, an ungulate group located there will graze. The group will continue to graze throughout the day until it reaches its maximum daily intake, which is a function of the initial body weight of the ungulates at the beginning of winter.

Two negative feedback terms, ranging from 0 to 1, are included in the model to regulate ungulate foraging. A feedback value of 1, indicating no limitation, allows the group the maximum forage intake, while a feedback value of 0 allows no forage intake. The lower of the two feedback terms is allowed to operate on the daily forage intake.

The first feedback term, `ynpr`, represents the effect of a reduction in the amount of available forage on the rate of forage intake. As animals graze at a grid cell, this feedback value is decreased to represent a smaller amount of forage. The biomass at a grid cell must be above a certain level, the refugium level, otherwise the ungulates will not be able to graze.

The second feedback term, `fbsm`, represents the effect of snow on the ability of an animal to obtain forage. Because animals generally cannot forage in snow that exceeds their brisket height, snow depth may limit foraging. Snow density can also be a problem since shallow dense snow may preclude foraging. Thus, snow-water equivalent, which integrates both depth and density, is used as a feedback term.

1.2.6. Search and Movement Rules

Although there are several foraging rules that can be used to reflect animal movement (Turner et al., 1993), the algorithm used in NOYELP to simulate search and movement for the animal groups is simple and somewhat conservative (i.e., not necessarily the most efficient rule for finding available forage). If a group is located on a grid cell with available forage, the group will graze. Otherwise, the group will search in concentric rings, up to a radius equal to the maximum moving distance, for another grid cell with available forage. Maximum moving distance is a function of ungulate type and snow conditions. Available forage is determined by the worse of two feedback terms. One is based on absolute abundance of forage (`ynpr`) and the other (`fbsm`) is based on snow conditions on the grid cell. The feedback term with the lower value, the one that will determine movement, is stored in the PMI variable (preferential moving index). The ungulate group will move to a grid cell, within a concentric ring, with the highest feedback term greater than 0.1. The threshold value of 0.1 was empirically derived. If multiple grid cells in a concentric ring have the same feedback value, the ungulate will randomly choose one of the grid cells with the highest value. The same movement sequence is used if the animal group is located on a pixel without available forage. If no available forage is found within the search radius, the animal group is placed randomly at its maximum moving distance.

Figure 2 illustrates the search pattern of an animal group with a maximum moving distance of 3 pixels, and the selection of a new grid cell. The animal group will search the dark gray grid cells first. The number in a grid cell represents a PMI value at that pixel. Because none of the PMI values in the first concentric ring are above the threshold of 0.1, the group continues searching in the next concentric ring (medium gray). Many grid cells in this ring have a PMI value above 0.1, thus the animal group selects the grid cell with a PMI value of 0.8, which is the highest value in that concentric ring. If the group had not found a suitable grid cell in this concentric ring, it would have continued the search until it had reached its maximum moving distance (black border).

The ungulate group will continue to search each day until it has met one of two conditions: it has reached its maximum moving distance, or it has reached its maximum daily intake. If the group has reached its maximum moving distance for the day, the group is not allowed to forage any more that day. The next ungulate group is then located on the landscape and the procedure repeated until all ungulate groups have completed foraging for the day. This algorithm, of course, does not simulate interactions between animal groups. Such interactions usually do not have an strong impacts on annual and semi-annual (as opposed to daily) survival rates and the frequency/proportion of habitat use, which were the desired output statistics from the original NOYELP model (Turner et al., 1993).

1.2.7. Ungulate Energetics

Daily energy balance, the difference between energy gain and cost, are computed at the end of each day for each ungulate group. Energy gain is a function of the total amount of forage consumed each day, while energy cost is dependent on maintenance energy and energy required for travel. Maintenance energy includes the cost of all the animal's daily activities: standing, resting, grazing, ruminating, and thermo regulating. The cost of travel is based on actual distance traveled and snow conditions at the grid cell the animal is leaving.

After the daily energy balance is computed, the body weight of the ungulates is computed. If the energy balance is positive, the ungulates do not gain weight, otherwise they lose weight.

1.2.8. Program Flow

The major subroutines defining the program flow of the NOYELP model (see Uziel, 1994) are illustrated in Figure 3. Static data layers and ungulate specifics are the inputs to subroutine `input`, while dynamic data layers and ungulate initial conditions are initialized in `initial`. Subroutine `ssnow` updates snow conditions every 3 days. Subroutines `ungfind`, `graze`, `move`, and `energet` are all involved in the movement sequence.

Subroutine `ungfind` locates an animal group on the landscape and allows it to begin the movement sequence discussed earlier in this section. Subroutine `graze` allows the animal to forage on a grid cell and `energet` records the daily energy balance statistics. `patch` is an optional subroutine that identifies clusters of high and low biomass patches. Although it not used in PNOYELP development, subroutine `patch` can be used to identify clusters of high and low biomass patches (Minser,

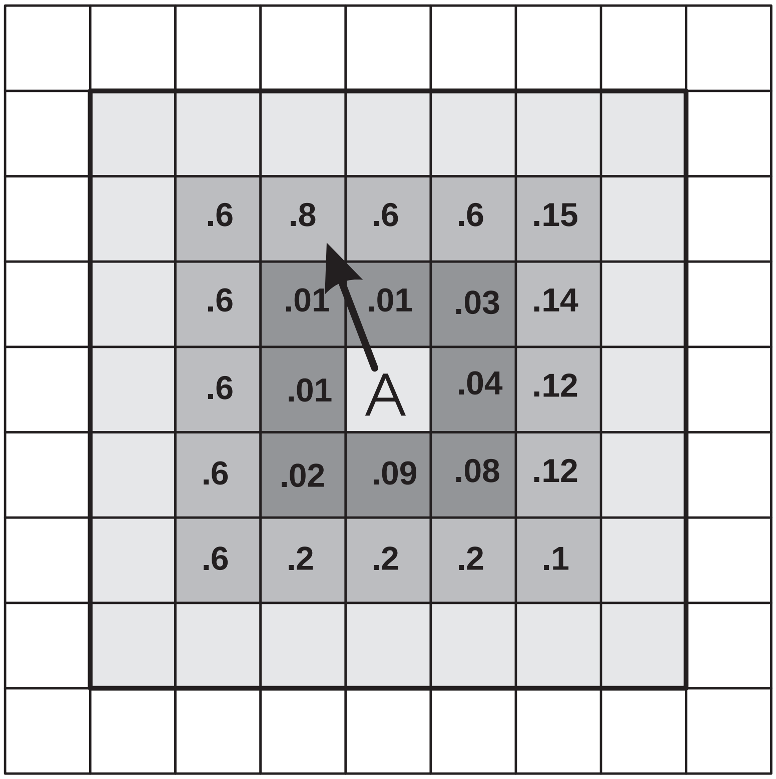

Figure 2. Search and movement patterns of animal groups in NOYELP.

1994). Subroutine `outfile` is another optional subroutine which will write dynamic data layers (i.e., arrays storing available biomass, ungulate location) to a diskfile.

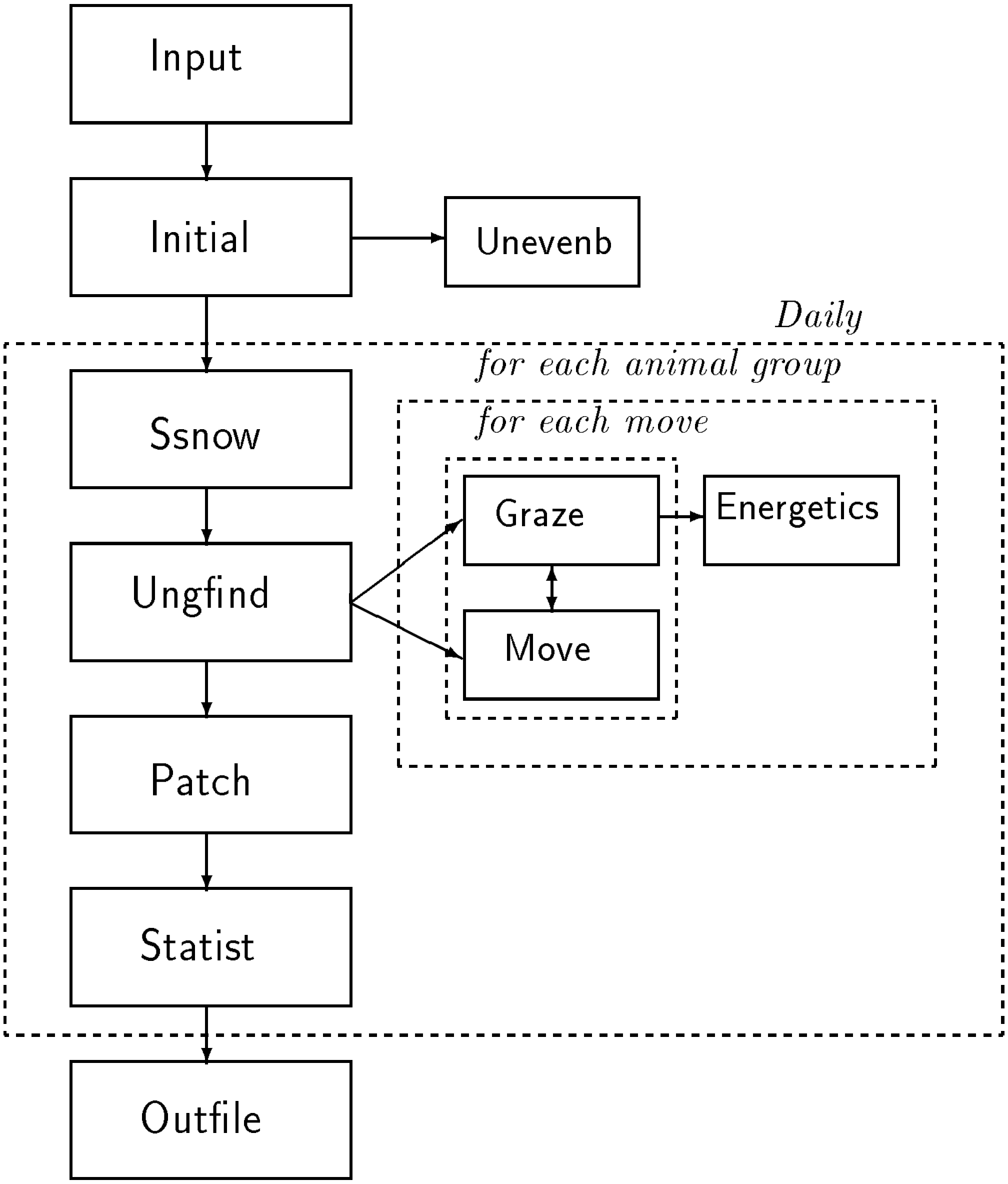

Figure 3. NOYELP model program flow by subroutine.

2. Parallel Model

The parallel model (PNOYELP) which exploits the the 32 processors of the Thinking Machines CM-5 is not a simple translation of the sequential NOYELP model. This section discusses the necessary modifications and the justification for the changes as well as new assumptions used to create the parallel model (Figure 4).

2.1. Host/Node Design

As discussed in Uziel (1994), the bulk of time in NOYELP is spent in the **`main`** program and subroutines **`move`** and **`ungfind`**. Since the move sequence for the animal groups includes the **`move`** and **`ungfind`** subroutines, all other subroutines involved in the move sequence were transferred to the node program of PNOYELP. These transferred subroutines included **`ungfind`**, **`graze`**, **`move`**, and **`energetics`**. The **`ssnow`** subroutine was also chosen to run on the nodes to avoid having the host update the nodes every 3 days with new snow conditions. Most of the **`main`** program was also implemented in the node portion of the program because it did not have to run sequentially.

Since over 93% of the original sequential program could be executed in parallel (see Uziel, 1994), Amdahl's law (Hwang, 1993) suggests that the maximum speedup possible on 32 processors of the CM-5 would be

$$S_{32} = \frac{1}{0.07 + \frac{.93}{32}} \simeq 10.1,$$

assuming a fixed work load. In order to attain a fixed work load in both models, PNOYELP requires survival rates[2] similar to those of the sequential NOYELP model. Figure 4 illustrates the program flow of the parallel model.

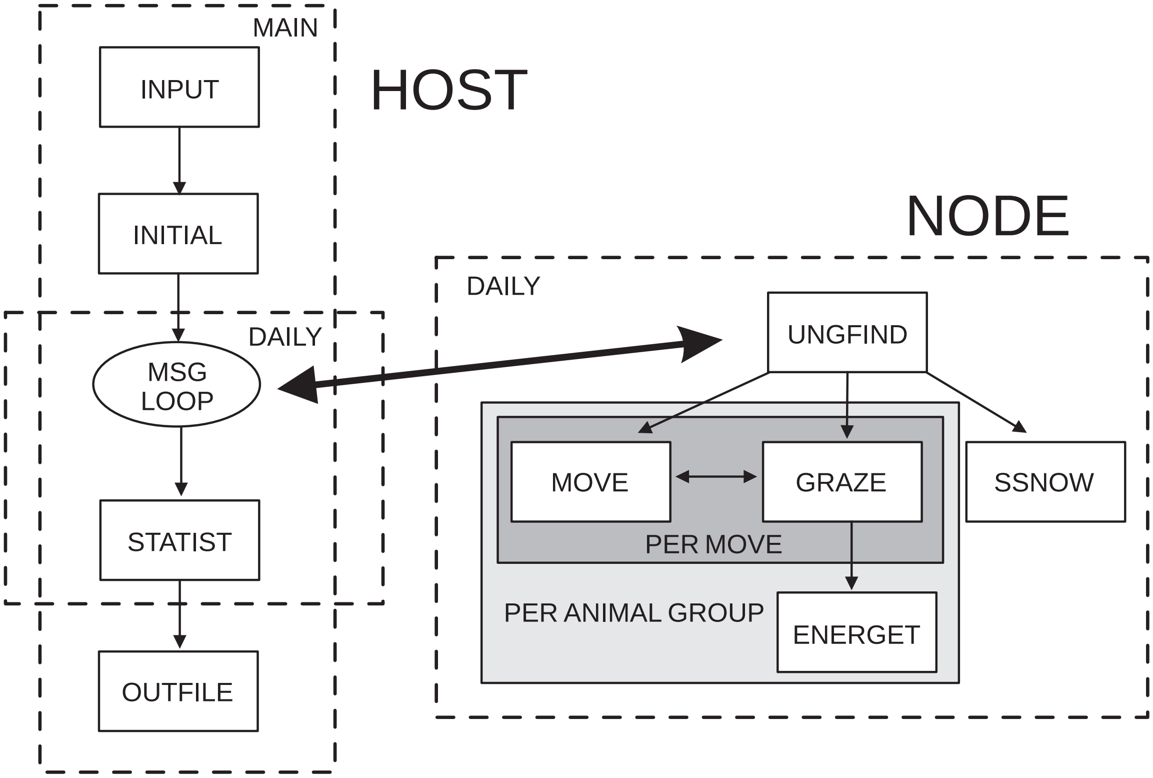

Figure 4. PNOYELP program flow by subroutine.

2.2. Partitioning of the Landscape

The landscape in PNOYELP was partitioned across the 32 processors nodes of the CM-5 in order to balance the work load on each processor. Initially, a regular grid was used to partition the landscape among the processors (Figure 5). Such a regular grid, however, forces an obvious imbalance in the amount of work across the PNs as some processors are not allocated any portion of the study area.

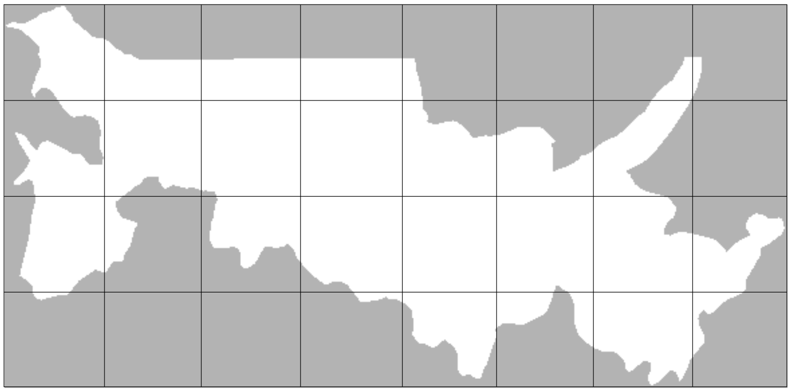

Figure 5. NOYELP landscape with a regular grid partitioning scheme.

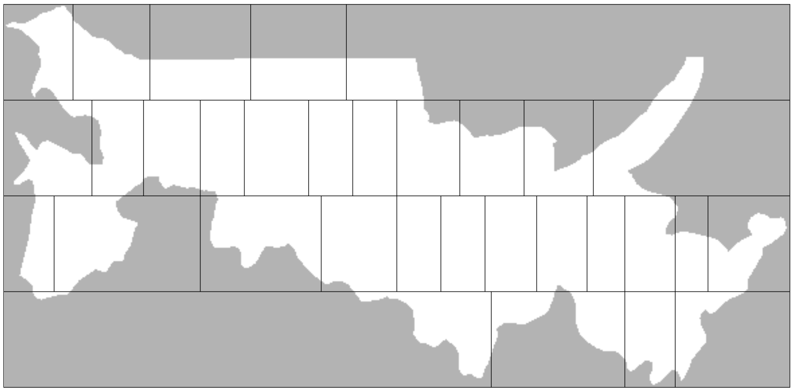

Figure 6. NOYELP landscape with an irregular grid partitioning scheme.

The regular grid was then replaced with an irregular grid (Figure 6) for two reasons. One was to ensure a more balanced computational load across the PNs, and the second was to create a suitable partition on which the animal groups would find enough food and thus remain on the processor for as many days as possible (i.e., animal group localization). In order to derive a suitable irregular grid, the amount of available biomass on each partition was analyzed to determine how long the animal groups would remain on a given partition. The number of animals per processor was also taken into account since time spent in the sequential NOYELP model is very dependent on the number of animals (Figure 7).

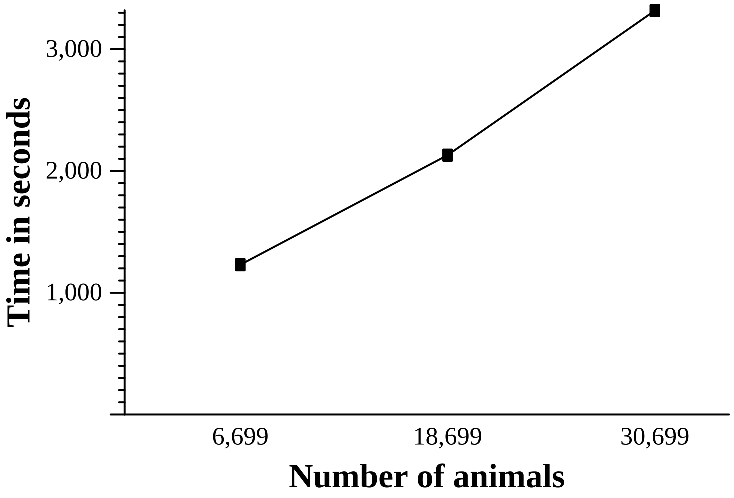

Figure 7. CPU time (in seconds) of the original sequential NOYELP model based on different animal totals.

Another design criteria for the irregular grid was potential dynamic resizing of the grid during the 180-day simulation. Without this consideration, a more equitable partitioned grid could have been created without row boundaries. Although no dynamic repartitioning of the landscape was actually used, a strategy for refining the irregular grid in Figure 6 is briefly discussed in Section 4.

2.3. Node Initialization

The host program of PNOYELP begins much like the sequential NOYELP model. The data layers (elevation, habitat, aspect, slope, snow, and initial forage) are initialized along with ungulate initial conditions as discussed in Section 1. The host must then broadcast this information to the 32 PNs so each will be able to determine which animal groups and what portion of landscape they are responsible for. The information is also used to update snow conditions and to allow the animal groups to move and graze over the landscape.

The node program then determines its nearest-neighbor partitions of the landscape based on the irregular grid. A nearest-neighbor is any partition that touches one of a partition's borders. Figure 8 illustrates processor A with its nearest neighbors (labeled with a B). This information is necessary so animal groups have access to neighboring partitions which they can possibly move to. Since the nearest-neighbor partitions are determined during the simulation based on the initial irregular grid, the grid could be repartitioned during the simulation and nearest neighbors recalculated (discussed in Section 4).

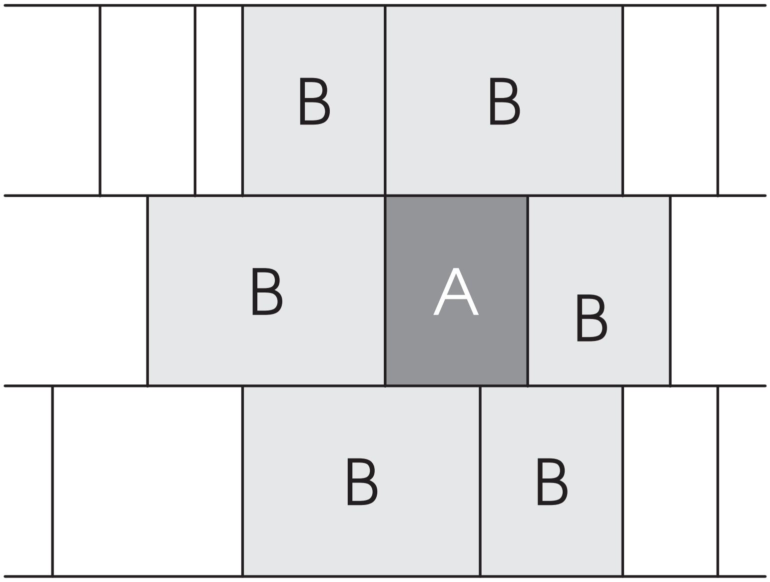

Figure 8. Partition A with nearest-neighbor partitions (B).

The last step in the node initialization requires the nodes to determine which animal groups it owns. This is done by placing a 1 in the array `nung_yes`, which has an entry for each group. This data structure was used to maintain the random ordering of the animal groups while moving between processors. Each processor is responsible for recording and updating vital statistics (daily energy balance) for animal groups resident on its portion of the landscape and initiating their movement sequence.

2.4. Animal Movement

Once node initialization is complete, the animals are ready to begin the 180-day movement sequence. The movement sequence in PNOYELP is very similar to that of the sequential model (NOYELP). The random moving order, determined by the *animal group number*, is preserved on each processor. The animal group with the lowest group number is located on each processor and allowed to start the move and graze sequence. The group will search the landscape on its processor within its maximum moving distance for a pixel with a PMI value greater than 0.1 (as discussed in Section 1.2.6). Unlike the sequential NOYELP model where the group continues its move/graze sequence until it has met one of two conditions, PNOYELP will discontinue the move/graze sequence if one of three conditions is met: it has reached its maximum daily intake, it has reached its maximum moving distance, or it cannot find a suitable grid cell within its maximum moving distance on its processor. Since the validation outputs of both the PNOYELP and NOYELP models are trends in the semi-annual (as opposed to daily) energy balance statistics, the third condition was judged by experts (Turner et al., 1994) as a reasonable modification to the sequential move/graze sequence used in NOYELP.

2.4.1. Determination of New Processor

If the animal group searches its processor within its maximum moving distance and cannot find a suitable pixel (one with a PMI value greater than 0.1), the animal group determines if there is nearest-neighbor processor it can move to. The new processor must be within its maximum moving distance and have a PMI average greater than 0.1 (discussed in the next two sections). If both conditions are satisfied, the animal group is allowed to move to the other processor to continue its movement sequence. Otherwise, the animal group is placed randomly at its maximum moving distance on its processor.

2.4.2. PMI Average

The PMI average, used to determine if an animal group is allowed to move to another processor, is calculated at the beginning of each day. The PMI value at each habitat pixel (a pixel in the study area) is summed and divided by the total number of habitat pixels to attain a PMI arithmetic mean for each partition. Other methods can be used, of course, to determine an average PMI value (to be discussed in Section 3).

The average on each processor is then broadcast to every other processor, so each processor will have a copy of every other processor's PMI average. Even though each processor only needs to know the PMI averages of its nearest neighbors, a broadcast to all processors was used because of its superior performance over point-to-point communication on the CM-5 (Thinking Machines Corporation, 1993).

Using a PMI average can significantly reduce communication time (Comiskey, 1993), and still facilitate similar foraging activity for elk and bison. Three data layers or globally-shared arrays (`fbsm`, `ynpr`, and `biotot`) are needed to calculate the PMI value at each pixel. The `biotot` data layer gives the absolute amount of forage available at a given grid cell. The `fbsm` data layer is the feedback term based on snow, and `ynpr` is the feedback term based on available forage as discussed in Section 1. For the `ynpr` and `biotot` data layers, the value corresponding to the pixel the group has chosen to graze on will change, so that if these data layers were exchanged between nearest neighbors once a day, they would be invalid as soon as a group grazed on another pixel. Also, in order to exchange the data layers between nearest neighbors once a day, 171 messages would need to be sent and received between nodes. In a worst case scenario, the node with the largest partition of the landscape would be required to send 2.5 million bytes of information in order to update its neighbors once per day.

2.4.3. Nondeployment of Guard Strips

The use of overlapping landscape partition boundaries or *guard strips* for sharing the `ynpr` and `biotot` data layers between neighboring processors was not considered due to the invalidation problem mentioned above and to the variations in maximum moving distances among animal groups. Although the snow data layer (`fbsm`) would only need to be shared every 3 days, the other two layers (`ynpr` and `biotot`) need to be shared/updated after each animal group moves. Once an animal group has grazed at a particular pixel within a guard strip, the data values on all shared `ynpr` and `biotot` layers (or arrays) are invalid. Any animal group on a processor sharing the guard strip would have either foraged using invalid information or be forced to wait (in lock-step) until all animal groups on every processor within its maximum moving distance had moved before completing its own daily movement (i.e., the sequential move/graze sequence in the NOYELP model). The invalidation phenomenon would especially occur for cases in which the guard strip contains an unusually high amount of biomass and is within the maximum moving distance of several animal groups (on several processors). Since the maximum moving distance for elk (over the 180-day simulation) is 40 pixels, as many as 10 processors (see Figure 6) could easily share data layers from the same guard strip. The current PNOYELP model without guard strips guarantees that animal groups only forage actual biomass available on a given pixel during their daily movement sequence.

2.4.4. Moving to New Processor

Once a group has decided to occupy a new processor's partition of the landscape, the movement sequence for that group is halted and variables needed to continue movement on the next processor are saved. The animal group number is saved so that the destination processor will know when to allow the animal to begin its move/graze sequence for the day. The current bodyweight is saved for the daily energy balance calculations at the end of the day. The current row and column number will be needed to determine placement of the animal group on the destination processor. The amount of food the group has grazed and the distance traveled so far that day is saved so that the animal group can continue its move/graze sequence until it reaches its maximum moving distance or its maximum daily intake. The snow density coefficient is saved to calculate maximum moving distance for the day and for the daily energy balance calculations. If a cow group has a calf group, all corresponding calf information is also saved for the destination processor. The sending processor also includes its processor number so the receiving processor will be able to correctly place the animal group on its landscape partition.

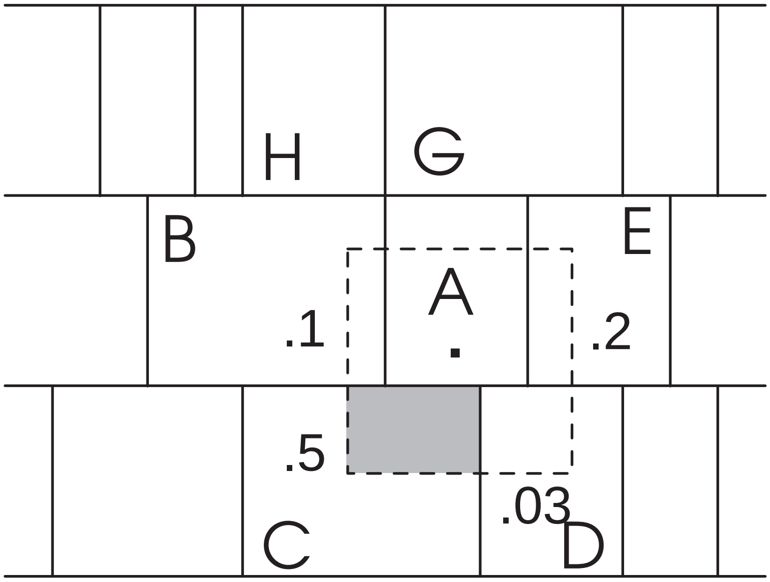

Figure 9. Animal group on processor A determines a new processor to move to and is placed within its maximum moving distance on the new processor (C).

The group searching for a remote processor's partition to occupy will determine which nearest-neighbor PNs are within its maximum moving distance for that day. If one or more neighbor PNs satisfies that condition, the group will check the PMI average of each of those processors and move to the neighboring PN with the greatest PMI average. Figure 9 illustrates an example of a group choosing a new processor to move to. The group located on processor A has a maximum moving distance defined by the dashed line square surrounding it. Since the group's maximum moving distance falls within the landscape partition owned by processor's B, C, D, and E, the group must determine which of these nearest neighbors has the largest PMI average (the number in each partition). Processor C's PMI average is above the suitability threshold (0.1) and is larger than the other processors' PMI averages, thus the group will be placed randomly within its maximum moving distance on processor C (shaded grey).

If several neighboring PNs within the maximum moving distance have the same average, the group will choose the destination PN randomly among the set of neighboring PNs. The group is then removed from the processor's ungulate location map and the list of groups it owns. If a cow group has a calf group associated with it, the calf group is also deleted from the ungulate location map and from the list of groups that processor owns.

The movement sequence for the remaining ungulate groups continues until the end of the day. Then, a message is sent to each nearest-neighbor PN containing either an array with ungulate information or a null message. The receiving processor accepts the message and checks for ungulate information or a null message. If there is group information, the message is unpacked and data layers and ungulate information are updated on the destination processor, so the animal group can continue its move/graze sequence. The animal group is then placed randomly within its maximum moving distance on the receiving processor and its move/graze sequence is continued for that day.

3. Validation and Performance

As mentioned in Section 1.2.6 and at the beginning of Section 2.4, the validation of PNOYELP is based on trends in semi-annual energy balance statistics. Daily ungulate location maps from the PNOYELP and NOYELP models may or may not be similar due to the different animal movement rules described in Section 2.4. This section briefly discusses the primary differences and similarities between the selected outputs from the parallel and sequential models, and the effects of using different PMI averages. Performance results and speed improvements obtained with all versions of the parallel model are also presented.

3.1. Comparison of Selected Outputs

Several daily energy balance statistics (see Uziel, 1994) can be used to record characteristics of the animal groups and landscape. Two critical daily energy balance statistics, *survival rate* and *daily travel distance* of elk cows, can be used to verify the entire model, and hence serve as the primary statistics for comparison of the NOYELP and PNOYELP models. These statistics were plotted and compared to determine if the animal groups were moving the same distance with a similar survival rate in both models.

3.1.1. Travel Distance and Survival Rates

The original PNOYELP version (AM) used an arithmetic mean to compute the PMI average on each processor (Section 1). Because NOYELP is a stochastic model, daily energy balances are calculated over 5 repetitions, each with a different random number seed. In Figures 10 and 11, similarities in both the daily travel distance and the survival rate of elk cows in the NOYELP and the PNOYELP models are illustrated. A slight increase in ungulate deaths in the parallel model is explained by the greater distance traveled (as seen about day 165 through day 180 in Figure 10), which is due to the fact that the groups search their current processor for suitable grazing pixels before they are allowed to move to another processor. Requiring an animal group to

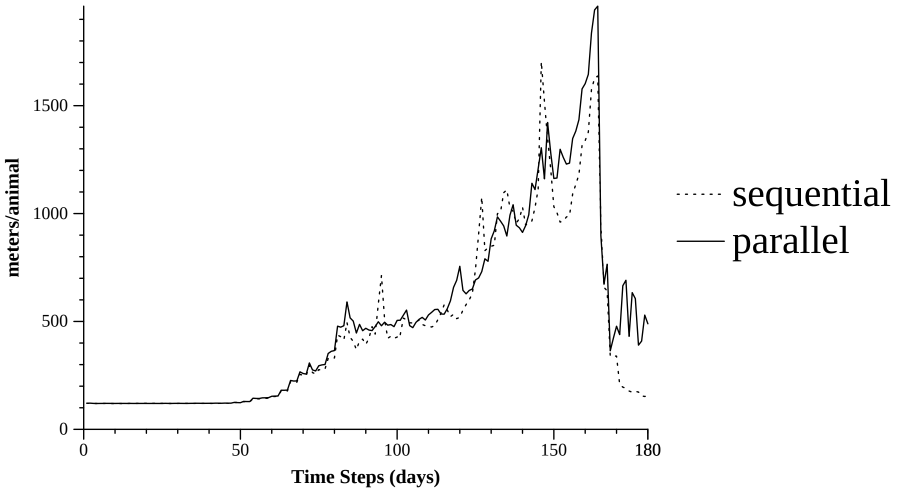

Figure 10. Daily travel distance of elk cows for 5 repetitions.

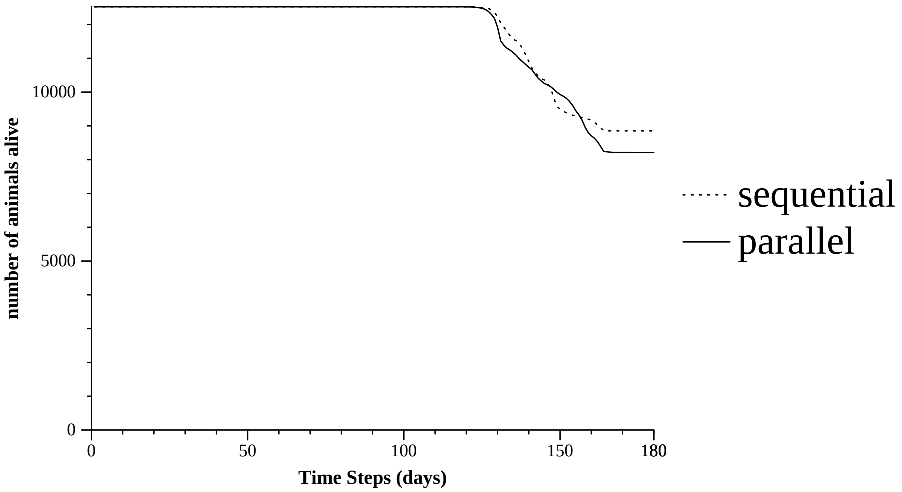

Figure 11. Daily survival of elk cows for 5 repetitions.

search its local partition before checking other processors may cause one or both of two events which can increase daily travel distance: (1) a group may find a suitable grid cell on its own processor, but at a further distance, or (2) a group might require more moves within its current processor's landscape partition in order to consume its maximum daily intake. Since the groups are moving further in the parallel model, they expend more energy and lose more weight, thus ungulate mortality increases. Even though there are differences between the models in the energy balance statistics, the differences are considerably small. The *daily survival rate* graph illustrates an acceptable range of 0% to 8% difference between the PNOYELP and NOYELP versions, while 0% to 25% is the range of differences in the *daily travel distance* graph up to day 165.

3.1.2. Ungulate Location Maps

The ungulate location maps, which show the location of ungulate groups across the PNOYELP and NOYELP landscapes, demonstrate trends in animal foraging based on animal movement rules. On day 1, the ungulate groups are placed randomly across the landscape (Section 1.2.4), as illustrated in Figures 12 and 13. In these figures, each white dot on the landscape represents between 1 and 5 groups per grid cell.

By day 120, the animal groups in both the sequential and parallel models have moved in the same direction and tend to converge to the same areas of the landscape (Figures 14 and 15). By day 155 (Figures 16 and 17), there are two noticeable differences between the PNOYELP and NOYELP ungulate location maps: (1) some processor partition definition is visible in the PNOYELP map because the groups search within their maximum moving distance on their local processor before attempting to move to a new processor, and (2) a large cluster of groups of animals has congregated in the northwestern portion of the landscape partition (assigned to PN 1). The density of animal groups on PN 1 is a result of the animal groups in PNOYELP moving to the same landscape partitions a few days later than their counterparts would in the sequential model. To verify that the groups in PNOYELP were traveling in the same direction as the groups in NOYELP, ungulate location maps were dumped every 5 days between day 135 and day 150. For either the sequential or parallel models, the animal groups tend to move in a clumped-fashion along the northern border towards the west and then move south along the western border. Producing equivalent ungulate location maps between the PNOYELP and NOYELP models was not an expected result in this research since daily movement patterns for even the sequential NOYELP model are not necessarily exact representations of elk and bison movement (see Section 1.2.6). The spatial effects of changes in animal movement rules, however, are useful in the interpretation of travel distances and survival rates for ungulates.

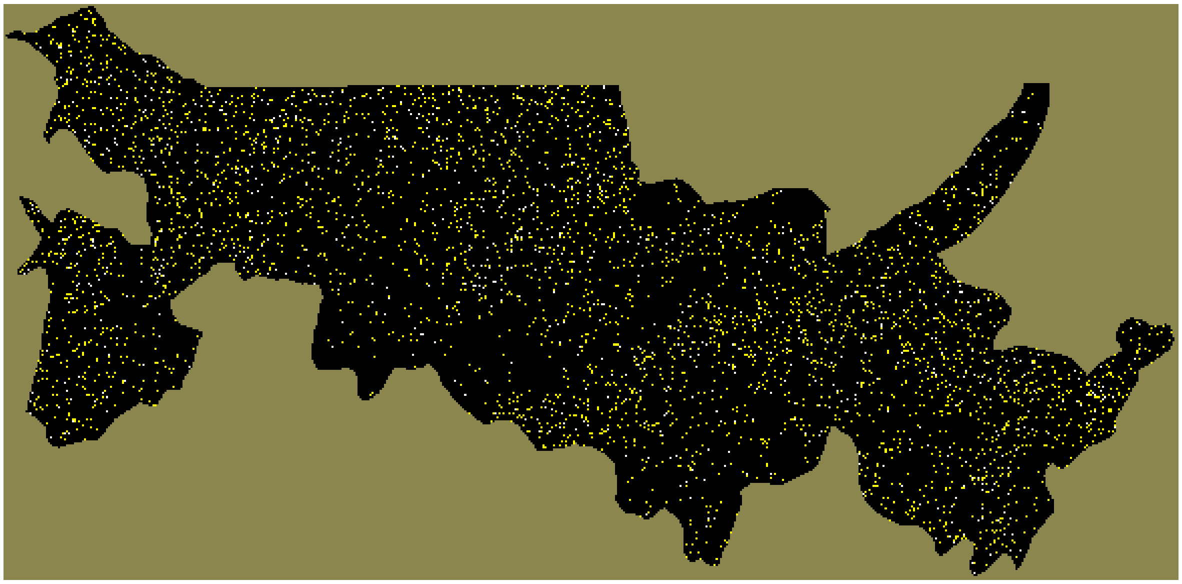

Figure 12. Sequential model ungulate location map for day 1.

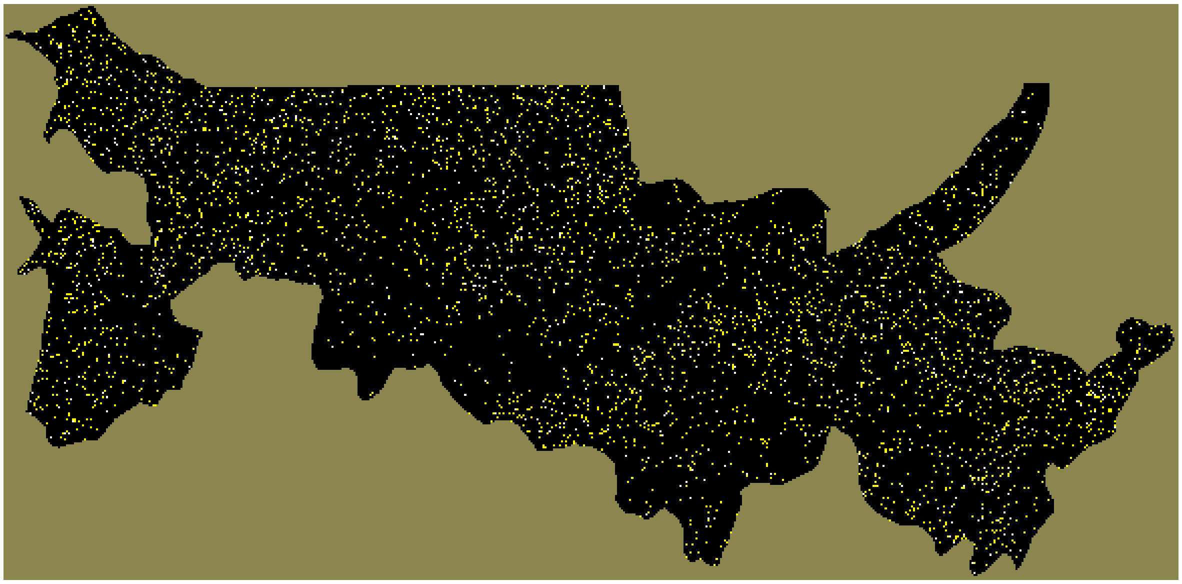

Figure 13. Parallel model ungulate location map for day 1.

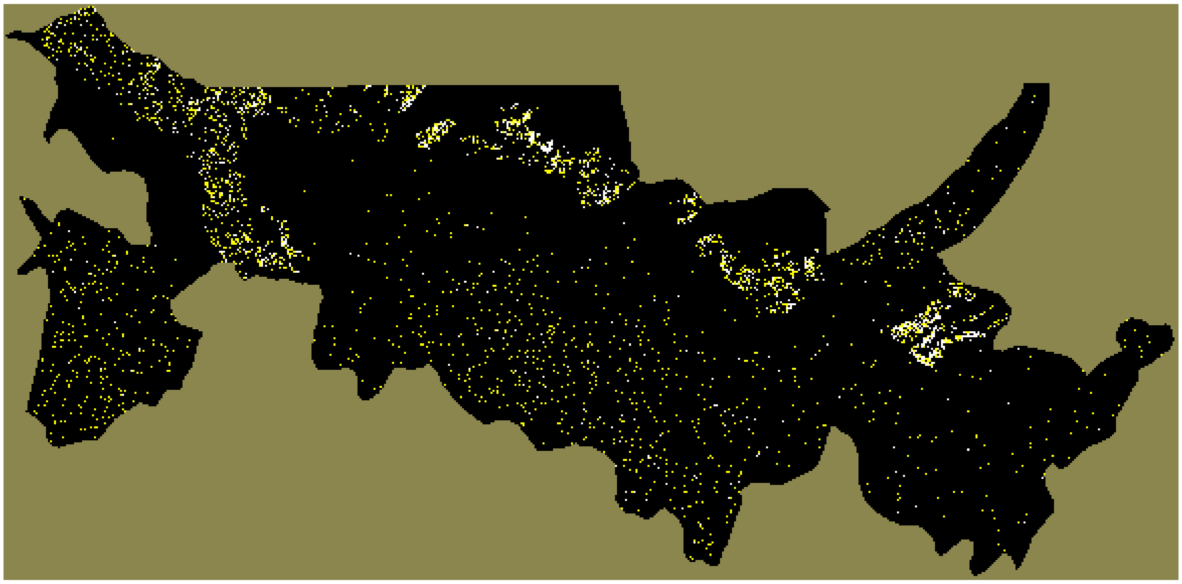

Figure 14. Sequential model ungulate location map for day 120.

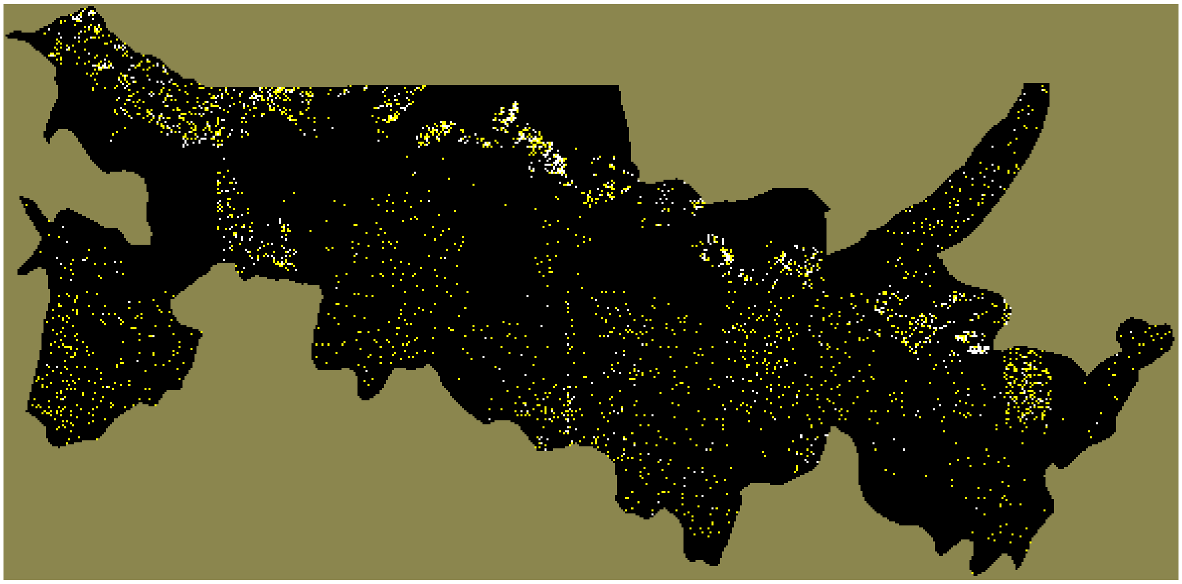

Figure 15. Parallel model ungulate location map for day 120.

Figure 16. Sequential model ungulate location map for day 155.

Figure 17. Parallel model ungulate location map for day 155.

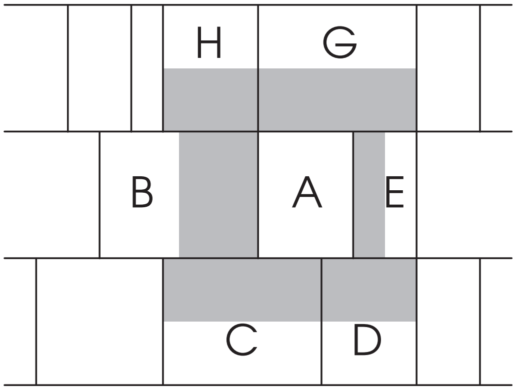

Figure 18. Partial averages compared by an animal group located on processor A.

3.2. Modified PMI Averages

Two modified versions of the original PNOYELP model were also implemented to study the effects of using different PMI averaging techniques on energy balance statistics and ungulate location maps. The *threshold* version of PNOYELP (TH) varied the PMI arithmetic mean threshold, i.e., the minimum PMI average (or mean) that a neighboring processor must have in order for an animal group to move there. This method allowed animal groups to move to another processor only if the receiving processor's PMI arithmetic mean was within a certain percentage of the sending processor's PMI arithmetic mean. For example, if the local processor's PMI arithmetic mean is 1.0 and the animal groups are only allowed to move to processors within 30% of the local processor's PMI arithmetic mean, the animal group attempting to move will only be allowed to move to a processor with a PMI arithmetic mean between 1.0 and 1.3. A range of percentages was used (0% to 100%) along with a range of PMI arithmetic mean thresholds (0.001 to 0.2) to determine suitability. However, this method showed no appreciable differences from the original AM version. In one instance of the TH version, the PMI threshold was set to 0.1, thus the neighboring partition had to have a PMI arithmetic mean of at least 0.1 to be considered suitable. In addition, the neighboring partition's PMI arithmetic mean had to be within 30% of the local PMI arithmetic mean between day 130 and day 150. This combination returned daily energy balance statistics and ungulate location maps most similar to the sequential NOYELP model. Because the groups could only move to processors with PMI averages within 30% of their own, they were prevented from moving directly to the partitions with the highest PMI average, and their progression was hence slower than that observed with NOYELP.

Another PNOYELP version (PA) computed a partial PMI average (arithmetic mean) over each half of the processor. Four averages were computed for each processor, one corresponding to each border of the partition (north, south, east,

Table 1. Wall-clock times (in seconds) for the NOYELP and PNOYELP models with a varying number of elk. Speed improvements of each parallel version over the original sequential version are also listed.

Parallel version	Number of elk	NOYELP	PNOYELP	Speed improvements
AM	6,000	1230	184	6.68
	18,000	2130	451	4.72
	30,000	3317	1168	2.84
TH	6,000	1230	184	6.68
	18,000	2130	388	5.49
	30,000	3317	1103	3.00
PA	6,000	1230	225	5.46
	18,000	2130	461	4.62
	30,000	3317	1146	2.89

and west). The partial PMI averages were then broadcast to the 32 PNs as in the original parallel version. An animal group that decided to move to a new processor compared the averages corresponding to the shared border. As illustrated in Figure 18, an animal group located on processor A checks the shaded portions of its nearest neighbors (B, C, D, E, G, and H) only if they are within the maximum moving distance. The only noticeable difference between the PA and AM version was that more time was spent each day sending the three other averages to the remote processors. The daily energy balance statistics and ungulate location maps remained very similar.

3.3. Performance of Parallel Versions

The sequential and parallel versions were compared based on speed improvement (i.e., how much faster the parallel version executed on the 32-processor CM-5 relative to the sequential NOYELP program running on a SPARCstation 10). Both models were compiled using the current Fortran-77 SPARCcompiler (`f77` version 2.01) using the `-O` option for optimization. The parallel model was also linked with `cmmd-ld` version 3.2, the CMMD link editor, to create the CM-5 executable files. Elapsed times recorded in seconds (wall-clock times) and speed improvements for PNOYELP are listed in Table 1. Each version of PNOYELP executed in substantially less time than NOYELP with a peak speed improvement of 6.7 times faster than the sequential version NOYELP running on a Sun SPARCStation 10. Considering a maximum possible speedup of 10.1 from Amdahl's law in Section 3.1, the speed improvements of the PNOYELP versions were quite reasonable. The speedup calculated by Amdahl's law required the animals not to move between processors, which is unrealistic and explains, at least in part, why the speed improvements of PNOYELP did not achieve the 10.1 maximum. The PNOYELP model incurred a substantial amount of communication and computation overhead moving animals between processors which is not accounted for in the sequential model. The difference in the number of animals used in a simulation greatly affected the speed improvements. As the number of animal groups increased, more animals were moving between processors, and the size of messages sent between processors grew proportionately. As would be expected, the PA version was somewhat slower due to the overhead of sending 3 additional PMI averages each day, (see Section 3.2).

3.4. Performance of PNs

A comparison of the computation time required on the PNs is an important determinate of how load-balanced the computation within the PNOYELP versions were. A balanced load ensures each PN will have about the same amount of computation while minimizing idle time. Only actual computation time is recorded on each PN in an attempt to measure the load (Figure 19). The variance in time between PNs is, for the most part, due to the number of animal groups on each processor, but sending and receiving animal groups each day incurred additional overhead. Figure 20 illustrates the total number of animal groups a processor was responsible for over 5 simulations of the AM version. To create a more balanced computational load across the PNs, each PN would need to remain responsible for an equal number of animal groups.

In order to measure idle time on each node, an attempt to obtain a histogram of message-passing times was made. The elapsed time spent sending messages on each node was very similar, since synchronous communication was used for all PNOYELP implementations. Unfortunately, CMMD timing routines do not offer a method of measuring idle times.

3.5. Recommendations

Among the different PNOYELP versions, the AM version provided the most consistent semi-annual energy gain statistics with reasonable speed improvements over the sequential NOYELP model. The ungulate location maps from the AM version represented the direction of animal movement very well, although some processor partition boundaries were visible at times. Although the other versions provided better speed improvements, their resulting survival rates were considerably lower due to the inability of animal groups to travel far enough to find suitable forage.

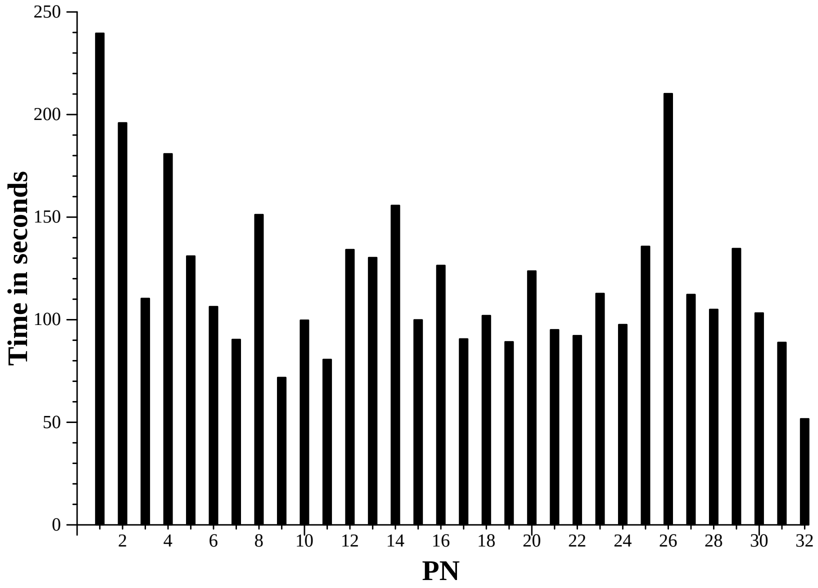

Figure 19. Total computation time on PNs for 5 repetitions of the AM version.

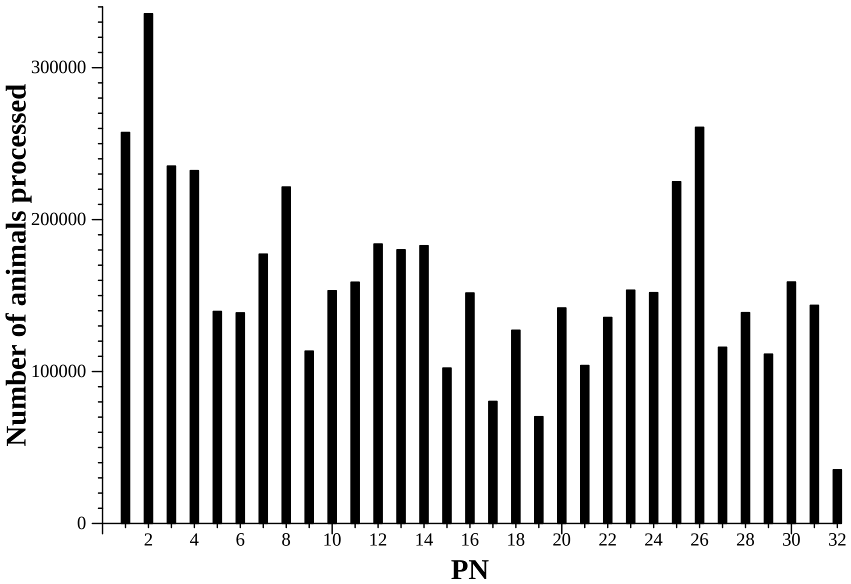

Figure 20. Total number of animals processed by each PN for 5 repetitions of the AM version.

4. Summary and Future Work

In this paper, a parallel animal migration model (PNOYELP) was implemented on the Thinking Machines CM-5. Results were very promising with reasonable speed improvements over a previous sequential model (NOYELP) and with acceptable semi-annual energy gain statistics. Modifications to the PNOYELP model using more assumptions of the original sequential model is possible, of course. However, such additions to the model could impair the speed improvements realized in the current PNOYELP model. An important modification discussed in Uziel (1994) is based on dynamic repartitioning of the landscape during a simulation. As the animal groups in the model progress through the 180-day move/graze cycle, they tend to converge in specific regions of the landscape. At that time, the processors that own the landscape partitions where the animal groups converge have more work than other processors because a majority of time spent in the model is in the move/graze sequence of each animal group (as discussed in Section 1). Resizing the partitions during the simulation to reflect the animal movement, should produce a more balanced workload among all processors.

Acknowledgement

This research was supported by the National Science Foundation under grant numbers NSF-ASC-92-03004 and NSF-CDA-9115428.

Notes

1. A *rain shadow* is a region of little rainfall on the lee slopes of mountains whose windward slopes receive the rain.
2. Survival rates within 10% of the sequential NOYELP model were acceptable output rates of the PNOYELP model.

References

1. BERRY, M., COMISKEY, J., AND MINSER, K. 1994. Parallel Analysis of Clusters in Landscape Ecology. *IEEE Computational Science and Engineering 1,* 2 (Summer), 24–38.
2. COMISKY, E. 1993. Data-Parallel Implementations of Map Analysis and Animal Movement for Landscape Ecology Models. Technical Report CS–93–207 (August), University of Tennessee, Knoxville, Tennessee.
3. CORPORATION, T. M. 1993. *Connection Machine CM-5 Technical Summary.* Cambridge, Massachusettes: Thinking Machines Corporation.
4. DESPAIN, D., RODMAN, A., SCHULLERY, P., AND SHOVIC, H. 1989. Burned Area Survey of Yellowstone National Park: The Fires of 1988. Technical report, Division of Research and Geographic Information Systems Laboratory, Yellowstone National Park, Wyoming.
5. DIAZ, H. 1979. Ninety-one Years of Weather Records at Yellowstone National Park, Wyoming, 1887-1997. Technical report, National Oceanic and Atmospheric Administration, Environmental Data and Information Service, National Climatic Center, Asheville, North Carolina.

6. Dirks, R. and Martner, B. 1982. The Climate of Yellowstone and Grand Teton National Parks. Occasional paper number 6, U.S. National Park Service, Washington, District of Columbia.
7. Gardner, R., O'Neill, R., and Turner, M. 1992. *Humans as Components of Ecosystems: Subtle Human Effects and the Ecology of Populated Areas.* S.T.A. Pickett and M.J. McDonnell, Springer-Verlag, New York.
8. Hargrove, W., Gardner, R., Turner, M., Romme, W., and Despain, D. 1993. Simulating Fire Patterns in Heterogeneous Landscapes: the analysis and interpretation of landscape heterogeneity. preprint.
9. Hwang, K. 1993. *Advanced Computer Architecture: Parallelism, Scalability, Programmability.* McGraw-Hill, Inc., New York, New York.
10. Turner, M., Wu, Y., Romme, W., and Wallace, L. 1993. A Landscape Simulation Model of Winter Foraging by Large Ungulates. *Ecological Modelling 69,* 3, 163–184.
11. Turner, M., Wu, Y., Wallace, L., Romme, W., and Brenkaert, A. 1994. Simulating Interactions Among Ungulates, Vegetation and Fire in Northern Yellowstone National Park During Winter. *Ecological Applications 4,* 3, 472–496.
12. Uziel, E. 1994. Parallel Animal Migration Models of Bison and Elk in Northern Yellowstone National Park, The University of Knoxville, Tennessee, Knoxville, TN.