Distributed Land-Cover Change Simulation [*]

M.W. Berry and K.S. Minser [†]

August 4, 1997

Abstract

Computer simulations are used in landscape ecology to simulate the effects of human land-use decisions on the environment. Such decisions are influenced by both ecological and socioeconomic factors which can be represented by spatially explicit multidisciplinary data. The Land-Use Change Analysis System (or **LUCAS**) was developed to study the effects of land-use on landscape structure in such areas as the Little Tennessee River Basin in western North Carolina and the Olympic Peninsula of Washington state. These effects include land-cover change and species habitat suitability. Using a geographic information system (GIS) to store, display and analyze map layers derived from remotely sensed images, census and ownership maps, topological maps, and output from econometric models, a parallel/distributed version of LUCAS (pLUCAS) was developed for simulations on a network of workstations. Targeting distributed computational environments reflects the resources available to most land-use planners, forestry personnel, and wildlife managers. A performance evaluation of two pLUCAS distributed models on an ATM-based network of 12 SUN Ultra-2 workstations is presented.

1 Introduction

One approach toward a better understanding of the the effects of human land-use decisions on the environment is to consider both ecological and socioeconomic factors. Such a multidisciplinary approach was taken by the Man and the Biosphere (MAB) project [1], whose goal was to analyze the environmental consequences of alternative land-use management scenarios in two different geographic regions: the Little Tennessee River Basin (LTRB) in North Carolina and the Olympic Peninsula in Washington State.

The MAB approach involved the integration of disciplines such as ecology, economics, sociology, and computer science to evaluate the impacts of land-use. Land-Use Change Analysis System (LUCAS) is a prototype computer application specifically designed to integrate the multidisciplinary data and to simulate the land-use policies prescribed by the integration model. The initial LUCAS prototype was implemented as an *object-oriented* C++ application to promote modularity. This modularity facilitates the addition of future software which might address the needs of different types of users. The creation of a distributed version [5] of LUCAS, Parallel LUCAS (pLUCAS), was motivated by the computational needs of real-time processing and extensions to larger regions. The first design of pLUCAS [5] utilized the Parallel Virtual Machine (PVM) [4] message-passing environment, and the (current) follow-up implementations are based on MPI [6]. While the major focus of this work involves a performance evaluation of the PVM and MPI implementations of pLUCAS, future research is needed in the development of a pLUCAS-based distributed geographic information system (GIS) for land-cover change simulations of larger study areas.

2 LUCAS Design

LUCAS provides a stochastic model for the future assessment of landscape change using historical maps of land cover. The initial design's modularity provides great flexibility for future modifications required by diverse users.

[*] This research has been supported by the Southeastern Appalachian Man and the Biosphere (SAMAB) Program under U.S. State Department Grant No. 1753-000574 and University of Washington Subcontract No. 392654, by the National Science Foundation under grants NSF-ASC-94-11394 and NSF-CDA-95-29459, and the USDA Forest Service under Contract No. 29-1286.

[†] Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN, 37996-1301, [berry,minser]@cs.utk.edu

2.1 Stochastic Modeling

The econometric model used in LUCAS is a dynamic, *stochastic* model primarily based on one random variable, namely land cover, and deals explicitly with time-variable interaction. The stochastic simulations enabled by LUCAS employ the statistical sampling of multiple replicates, i.e., repeated simulations of the same model. Figure 1 outlines the modular model used to develop the LUCAS prototype. Each module of the LUCAS model is briefly described in the following sections (see [1] for more details).

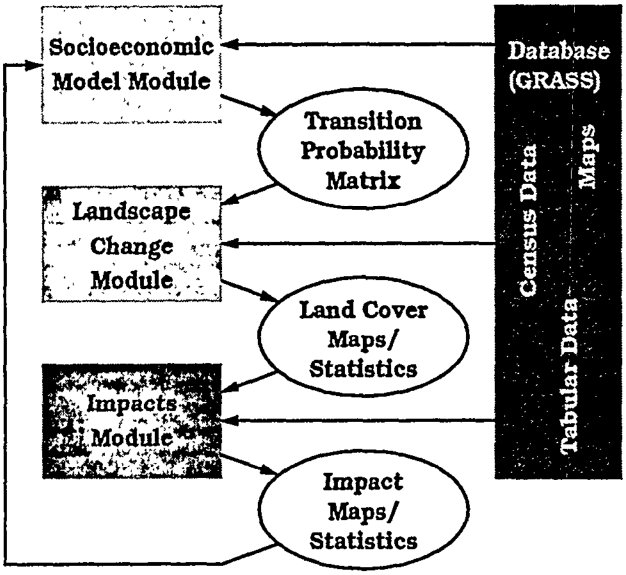

Figure 1: LUCAS Modular Design

2.2 Socioeconomic Module

Several discrete and continuous ecological and sociological variables were used empirically in calculating the probability of change in land cover: land-cover type (vegetation), slope, aspect, elevation, land ownership, population density, distance to nearest road, distance to nearest economic market center (town), and the age of trees. Each variable corresponds to a spatially explicit map layer stored in the GIS. A vector of all of these values for a given grid cell is called the *landscape condition label* [3]. An example landscape condition label (LCL) [5] is shown in Table 1.

Each element of the LCL $\vec{x} = (x_1, x_2, \ldots, x_8)^T$ is used to determine the probability of change using the multinomial logit equation [7, 1]

$$\Pr(i \to j) = \frac{\exp(\alpha_{i,j} + \vec{z}^T \vec{\beta}_j)}{1 + \sum\limits_{k \neq i}^{n} \exp(\alpha_{i,k} + \vec{z}^T \vec{\beta}_k)}, \qquad (1)$$

Table 1: Example Landscape Condition Label in the Hoh Watershed on the Olympic Peninsula

ℓ	x_ℓ	Definition	Attribute
1	2	Public Lands	Land ownership
2	75	75 years old	Tree age
3	1	Coniferous	Vegetation
4	20	20° incline	Slope
5	1	True	Steep slope > 17°
6	1500	1500 meters	Elevation
7	1000	1000 meters	Distance to town
8	21	1890 meters	Distance to road
9	2	0.002 people/acre	Population density

where n is the number of cover types, $\vec{z}$ is a 5×1 column vector composed of elements $x_4, \ldots, x_8$ of the LCL $\vec{x}$ in Table 1, $\vec{\beta}_j = (\beta_{1,j}, \beta_{2,j}, \ldots, \beta_{5,j})^T$ is a vector of logit coefficients, $\alpha_{i,j}$ is a scalar intercept, and $\Pr(i \to j)$ is the probability of coniferous land cover remaining the same ($i = x_3 = 1 = j$) at time $t+1$ or changing to another cover class (i.e., $j = 2, 3, 4$). The land ownership (x_1) determines which table of logit coefficients should be used and the tree age (x_2), used only for coniferous forest cover types, determines if the trees have aged sufficiently to be harvested, i.e., change to another cover type. The *null-transition* or probability of no land cover change is defined by Equation 2.

$$\Pr(i \to i) = \frac{1}{1 + \sum\limits_{k \neq i}^{n} \exp(\alpha_{i,k} + \vec{z}^T \vec{\beta}_k)}, \qquad (2)$$

where the symbols have the same meaning as in Equation (1). Example vectors of coefficients for the Hoh and LTRB Watersheds are available in [1] and [5]. Such coefficients and associated intercept values have been calculated empirically [1] from existing historical data stored in the GRASS database. The table of all probabilities generated by applying Equation (1) to all cover types is called the *transition probability matrix* (TPM), an example of which can be found in Table 2. If the TPM in Table 2 were used, for example, a random number from the closed interval [0, 1] less than 0.8725 would signify that the land cover would remain coniferous.

2.3 Landscape Change Module

The Landscape Change Module in Figure 1 is the heart of the LUCAS software. On input, this module accepts the multinomial logit coefficients generated in Socioeconomic Model Module, implements the actual landscape

Table 2: Example Transition Probability Matrix based on the example multinomial logit coefficients.

From Coniferous	Changing to	Probability
1 → 1	Coniferous	0.8725
1 → 2	Deciduous/Mixed	0.1186
1 → 3	Grassy/Brushy	1.886×10^{-3}
1 → 4	Unvegetated	6.989×10^{-3}

change, and produces new land cover maps and statistics as output. The first step in designing LUCAS was to develop the method to simulate one time step, a five year period, of landscape change over multiple replicates.

The computation of the pixel-based landscape transitions is relatively trivial because each grid cell changes independently. The transition probabilities from the initial cover type to all other cover types are calculated using Equation (1) and the value of the landscape condition label of a grid cell. A pseudorandom number is then drawn from a uniform distribution between 0 and 1. This number, in turn, determines the new land cover type for this grid cell via the calculated probabilities.

2.4 Impacts Module

The land cover maps produced by the Landscape Change Module are analyzed by the Impacts Module. This module may eventually determine the effect the changed landscape has on species, habitats, water quality,.or other environmental impacts. Currently LUCAS is designed to perform only species' habitat suitability analyses [1, 5]. The usual output from this particular module (not used for the results presented in Section 4) is a binary map; either a grid cell is suitable for a species or it is not.

3 Distributed Simulations

The motivation for the parallel and distributed implementation of LUCAS (pLUCAS) was to manage the multiple independent replicates required (for accuracy purposes) in the stochastic simulation of land-cover change. Parallelization is used so that each processor performs one complete simulation (replicate) of LUCAS exclusive of any other processor or process. In the performance tests presented in Section 4, 10 replicates of 20 timesteps each are performed for 4 different scenarios on the Hoh Watershed of the Olympic Peninsula. Along with these 40 tasks (4 scenarios × 10 replications) is a task for the initialization of any future impact modules that could be used so that a total of 41 independent tasks are scheduled.

The initial version of pLUCAS was implemented using PVM [5] and has now been modified to use the MPI message-passing software library [6]. All versions emulate a traditional *host/worker* model (see [2]) with some minor differences inherent to PVM and MPI. All pLUCAS runs using PVM and MPI were tested on a network of 12 Sun Microsystems Ultra 2 workstations, each containing two 167-Mhz UltraSPARC-1 processors under the Solaris 2.5.1 operating system (only 1 processor per workstation used due to contention for the single ATM port). Each workstation had 256-Mbytes of memory and two 2.1-Gbyte internal disks. Peak performance of one UltraSPARC-1 processor is about 126 Mflops (millions of floating-point operations per second). The workstations were connected by both a 10 Mbps Ethernet interface and 155 Mbps ATM sbus adapter so that performance results (recorded in elapsed wall-clock time) could be obtained with two different network latencies.

3.1 PVM Version

The PVM (version 3.3.10) implementation of pLUCAS allows the host process to assign tasks to worker processors by spawning a worker process onto a specific machine. The host maintains a queue of tasks to be scheduled, tasks completed, and available workers (machines). After all tasks are completed, the host spawns new processes on all machines to send data accumulated from their previous tasks back to the machine owning the host process. The host process collects the data, assembles it, and writes the results to files stored on a machine external to the ATM-connected network. The host processor is allowed to spawn a worker process to itself. Thus, the host machine will have one host and one worker process assigned to it. All other machines will have only one worker process at a time.

3.2 MPI Versions

To incorporate MPI (version 1.0.13) into the pLUCAS software, major code revisions were necessary due to the lack of process spawning with MPI. A traditional *host/worker* model was implemented using a top-level `if-then-else` construct to select the appropriate set of instructions for each process type. Two different pLUCAS versions using MPI have been developed. The first method, referred to as MPI(1), does not allow a worker process to coexist on the same machine as the host process. The second MPI version, referred to as MPI(2), is a better emulation of the PVM approach which allows for 1 worker process to be assigned to the host process machine.

4 Performance Results

Experiments conducted on the network of Sun Ultra 2 workstations involved 3 runs each with 1, 2, 4, 8, and 12 workstations to compute 10 replicates of 20 time steps for each of four historical, pixel-based scenarios of the Hoh Watershed on the Olympic Peninsula. Results reported for a single Sun Ultra 2 workstation reflect the use of the serial LUCAS implementation (i.e., no message-passing overhead).

For both PVM and MPI, the ATM (155 Mbps) network was slightly faster than the Ethernet (10 Mbps) network, but the difference was not significant. The fact that the only message-passing in pLUCAS occurs between the host and the workers (none between workers) might account for the small improvement of ATM over Ethernet. PVM had the best performance over either method of MPI with MPI(2) performing most similar to PVM. However, as more machines are added to the network, MPI(2) began losing its advantage over MPI(1) (see Figure 2).

Serial time for LUCAS on a single machine was 28.27 minutes. With 12 machines, the PVM implementation completed in 3.61 minutes yielding a speedup of 7.83. For the same number of machines, MPI(1) finished in 3.86 minutes followed by MPI(2) in 3.87 minutes with speedups of 7.32 and 7.30, respectively. With a 4-node machine, MPI(2) out-performed MPI(1) with speedups of 3.29 and 2.68, respectively. For an 8-node machine, MPI(2) yielded a speedup of 5.54, but MPI(1) maintained a speedup of 5.47. Finally, increasing the machine to 12 nodes resulted in MPI(1) and MPI(2) showing similar performances with MPI(1) actually being slightly faster than MPI(2).

The faster deteriorating performance of MPI(2) compared to that of MPI(1) can be attributed to process contention. Recall that for the MPI(1) scheme, the host process is scheduled on a dedicated processor and does not compete with any worker process. As more machines are included in the network, more message-passing demands are required of the host. In the MPI(2) scheme, the host machine has a host process as well as a worker process competing for the same port for message-passing. As the number of messages increase with added machines, the network contention becomes aggravated for MPI(2).

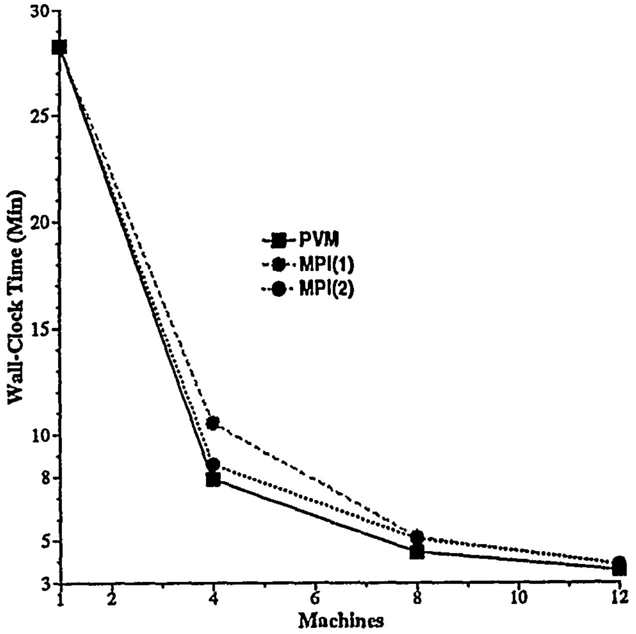

Figure 2: Timing comparisons of pLUCAS versions using an ATM-connected network.

References

[1] M. W. Berry, R. O. Flamm, B. C. Hazen, and R. L. MacIntyre. Lucas: A System for Modeling Land-Use Change. *IEEE Computational Science and Engineering*, 3(1):24–35, June 1996.

[2] M. W. Berry and K. S. Minser. Distributed Land-Cover Change Simulation Using PVM and MPI. Technical Report CS–97–356, University of Tennessee, Knoxville, February 1997.

[3] R. O. Flamm and M. G. Turner. Alternative model formulations for a stochastic simulation of landscape change. *Landscape Ecology*, 9(1):37–46, 1994.

[4] A. Geist et al. *PVM: Parallel Virtual Machine A User's Guide and Tutorial for Networked Parallel Computing.* MIT Press, Cambridge, MA, 1994.

[5] B. C. Hazen and M. W. Berry. The Simulation of Land-Cover Change Using a Distributed Computing Environment. *Simulation Practice and Theory,* 1997. In Press.

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

[7] D. N. Wear and R. O. Flamm. Public and private forest disturbance regimes in the southern Appalachians. *Natural Resource Modeling,* 7(4):379–397, Fall 1993.