Large-scale Land Use Optimization by Enhancing a Scalable Parallel Genetic Algorithm Library

[Extended Abstract]

Yan Y. Liu[12], Mengyu Guo[13], Shaowen Wang[12]
[1]CyberInfrastructure and Geospatial Information Laboratory (CIGI)
[2]National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign, Urbana, Illinois, USA
[3]Department of Industrial Engineering, Tsinghua University, Beijing, China
{yanliu, mengyu, shaowen}@illinois.edu

ABSTRACT

Optimization algorithms are often employed in spatial analysis and modeling to provide adaptive mechanisms at both individual and collective levels to enable decision-makers for the search of optimal solutions with respect to single/multiple objectives and constraints imposed by spatial configurations. This research aims to solve large-scale agricultural land use optimization problems by exploiting massive parallel computing resources provided by supercomputers such as those in XSEDE. The optimization of agricultural land use patterns finds an optimal assignment of crops (e.g., food and biofuel crops) on land parcels of a specified study area that maximizes the total yield and satisfies various competing constraints. These constraints often consider spatial factors such as contiguity and ownership, climate and land management factors (e.g., soil, precipitation, light, temperature, and ozone) and their effects on the productivity, suitability, and cost of assigning a crop on a land parcel. We have formulated the land use optimization problem as a classic combinatorial optimization problem - Generalized Assignment Problem (GAP) [2]. GAP is a well-known NP-hard problem [3]. When a landscape includes tens of thousands of land parcels (e.g., Figure 1), finding an exact optimal solution is computationally intractable.

In our research, we develop a parallel heuristic algorithm by combining an attention to the idiosyncrasies of agricultural land use optimization problem with a scalable parallel genetic algorithm (PGA) [4] to produce near-optimal solutions through scalable and efficient PGA computation on a large number of processors. Our PGA parallelizes the GA computation by running a large number of PGA processes simultaneously, each process conducting independent GA computation with a migration strategy that exchanges solutions between any two directly connected PGA processes at regular intervals. On each PGA process, a set of solutions form a local population. Standard GA operators such as population initialization, selection, crossover, mutation, and replacement are tailored to facilitate the search for better land use patterns based on aforementioned spatial and social economic factors. The parallelism in PGA is straightforward and easily permits a large number of PGA processes to evolve independently by following different randomized search paths and exploring the solution space collectively through migration strategies [1]. Nonetheless, a significant challenge remains regarding how to devise PGAs that are able to scale to massively parallel computer architectures. Issues persist because 1) a common PGA design adopts synchronized migration, which becomes increasingly costly as a larger number of processors are involved in global synchronization in each iteration; and 2) asynchronous PGA design and associated performance evaluation are intricate since the stochastic nature of PGA results in computations that are not simply dependent on the problem size.

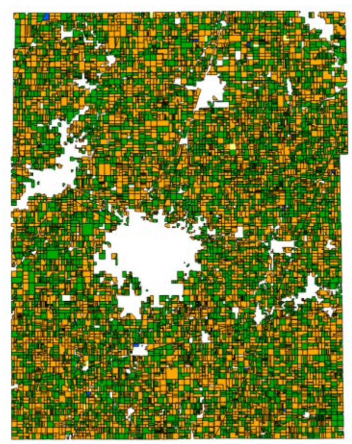

Figure 1. Land cover of Champaign County, Illinois, 2011. 11,957 land parcels. Green: corn; yellow: soy bean.

We addressed this challenge by developing an asynchronous PGA library that implements a scalable asynchronous migration strategy [4]. A suite of non-blocking migration operators (i.e., export, import, and inject) and buffer-based communications are developed to not only remove the costly global synchronization on migration operations, but also to allow for the overlapping of GA computation and migration communication. Buffer overflow issues caused by inter-process communications are resolved through algorithmic analysis. As a result, the relationship between the configuration of asynchronous PGA parameters (i.e., migration intervals, migration rate, and topology attributes) and buffer sizes is established based on the underlying message passing communication library and supercomputer interconnect

XSEDE'13, July 22-25, 2013, San Diego, California, USA.

characteristics to avoid buffer overflow issues at both system and application levels.

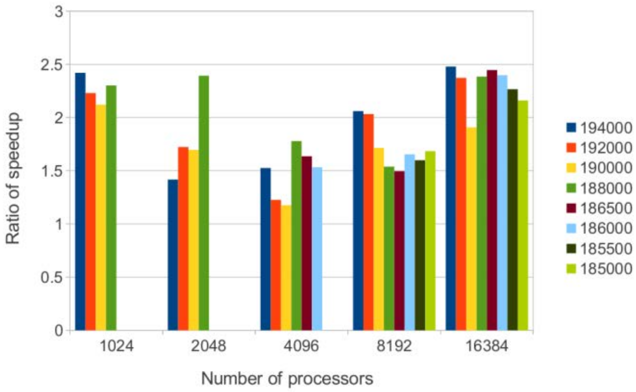

Figure 2. Ratio of speedup given different solution quality thresholds. Asynchronous *vs.* synchronous migration [4].

The scalability of our PGA library was evaluated by conducting strong and weak scaling experiments using up to 16,384 processor cores of the Ranger supercomputer at the Texas Advanced Computing Center. The design of these two experiments was tailored to evaluate the performance of asynchronous implementation of PGAs. Results indicated that our PGA library exhibited desirable speedups in the strong scaling experiment and impressive scalability to problem workload in the weak scaling experiment. Super-linear speedups were observed consistently as the number processor cores increased. The comparison between the asynchronous migration strategy and the corresponding synchronous implementation (Figure 2) is achieved by measuring the ratio of speedup (calculated by dividing the execution time of synchronous runs over asynchronous runs) at multiple solution quality thresholds. In all of the scenarios in which both our PGA and the synchronous version reached the specified solution quality thresholds, our PGA exhibited superior speedups. When using 16,384 processor cores, the speedup improvement was consistent across all of the solution quality thresholds. On average, the communication cost of our PGA was 15.5%, significantly lower than the synchronous version (54%). In the weak scaling experiment, the execution times of our PGA on 16,384 processors were consistently 60% less than the synchronous version at all of the solution quality thresholds as the global population size increased from 204,800 to 3,287,400 (Figure 3). The scalability enabled by the asynchronous migration strategy, in turn, greatly enhanced the problem-solving capabilities of the library to exploit massive computing power for solving large land use optimization instances.

Several enhancements to our PGA library are developed for the land use optimization problem-solving. Specific GA encoding mechanism and operators for efficient land use pattern search and fitness evaluation based on formulated spatial and social economic constraints are developed to improve the numerical performance of our PGA. Our PGA library is extended to adapt to supercomputers of hybrid architecture (e.g., Stampede cluster with mixed CPU, Intel Many Integrated Core (MIC), and GPU architecture). Specifically, the asynchronous migration strategy is enhanced and a runtime PGA parameter tuning function is developed for the library to be adaptive to dramatically increased heterogeneity among PGA processes on such supercomputers in order to achieve desirable scalability and reliability. Land use optimization results on the study area of Champaign County, Illinois are presented.

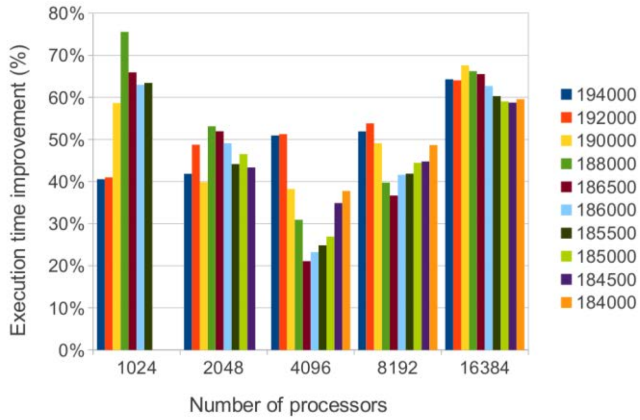

Figure 3. Weak scaling: reduction in execution time. Asynchronous *vs.* synchronous [4].

Categories and Subject Descriptors
C.2.4 [**Distributed Systems**]: Distributed Applications

General Terms
Management, Design, Security, Human Factors, Standardization

Keywords
Land Use Optimization, Heuristics, Genetic Algorithm, Parallel Computing, Scalability

ACKNOWLEDGEMENTS
This work is partially funded by the National Science Foundation (NSF) under Grant Number OCI-1047916. Computational experiments used the Extreme Science and Engineering Discovery Environment (XSEDE) (resource allocation Award Number SES090019), which is supported by the National Science Foundation Grant Number OCI-1053575. The authors thank the Champaign County GIS Consortium of the state of Illinois for providing the land parcel dataset and Tom Laue for the consulting service. We thank CIGI members Yanli Zhao, Myung-Hwa Hwang, and Julie Carlson for the help in processing land parcel and land cover datasets.

References

[1]. E. Alba and M. Tomassini. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 6 (5):443–462, 2002.

[2]. R. G. Cromley and D. M. Hanink. Coupling land use allocation models with raster GIS. *Journal of Geographical Systems*, 1:137-153, 1999.

[3]. M. L. Fisher, R. Jaikumar, and L. N. V. Wassenhove. A multiplier adjustment method for the generalized assignment problem. *Management Science* 32 (9): 1095–1103, 1986.

[4]. Y.Y. Liu and S. Wang. A Scalable Parallel Genetic Algorithm for the Generalized Assignment Problem. *Parallel Computing*, under review.