by means of a MapReduce processing model, which realizes effective parallelization and has strong fault tolerance.

In a previous study, Keco et al. proposed a method to parallelize GAs using MapReduce in a context similar to our proposed method. They adopted the island model and did not consider population migration; therefore, their method cannot secure sufficient population diversity to find a rigorous solution.

To solve this problem, we have proposed a method for population migration that is compatible with the MapReduce model.

We proposed a migration mechanism in Shuffle tasks. Shuffle tasks are the only tasks in which the MapReduce model allows modes to intercommunicate. To realize this, we have utilized the identification numbers of islands as keys to assign individuals to subpopulations. In addition, to reduce unnecessary network I/O in Shuffle tasks, we have also proposed a method to reduce the number of individuals during migrations.

We compared the performance of the proposed method with the original RCGA and its parallelized version without migration. The compared indices were the execution time and solution accuracy. To evaluate the effects of population reduction during migration, we compared the execution time for cases with and without population reduction. The results show a significant improvement in the solution accuracy and execution time.

In future, it will be necessary to discuss the optimum number of individuals reduced in Map tasks. Although we eliminated one-half of the individuals in each population, the ratio of elimination may be dependent upon the number of individuals. For example, assume that there are five subpopulations, each of which has one thousand individuals. Even though we eliminate one-half of these individuals in each population, many similar individuals could still be migrated to the same subpopulation. In this case, elimination of more individuals reduces the migration cost but will lead to the same solution.

References

[1] J. H. Holland, “Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence”, MIT press, Cambridge, MA, (1992).

[2] D. E. Goldberg, “Genetic algorithms in search, optimization, and machine learning”, Addison-Wesley Publishing Company Inc, (1989).

[3] M. Munetomo, “Genetic Algorithms, the theory and advanced technique” (Japanese), Morikita publishing Co. Ltd, (2008).

[4] E. Cantú-Paz, “A survey of parallel genetic algorithms”, Calculateurs Parallèles, Reseaux et Systems Repartis 10(2), pp. 141-171, (1998).

[5] J. Dean, S. Ghemawat, “MapReduce: simplified data processing on large clusters”, Communications of the ACM, 51(1), pp. 107-113, (2008).

[6] Message Passing Interface Forum, http://www.mpi-forum.org, (2014).

[7] L. G. Valiant, “A bridging model for parallel computation”, Communications of the ACM, 33(8), pp. 103-111, (1990).

[8] Apache Hadoop, http://hadoop.apache.org, (2014).

[9] T. White, “Hadoop: The definitive guide,” O’Reilly Media, Yahoo! Press, (2009).

[10] D. Keco, A. Subasi, “Parallelization of genetic algorithms using Hadoop Map/Reduce”, Southeast Europe Journal of Softcomputing, 1(2), pp. 56-59, (2012).

[11] H. Rasit, N. Erdogan, “Parallel Genetic Algorithm to Solve Traveling Salesman Problem on the MapReduce Framework using Hadoop Cluster”, JSCSE, 3(3), pp. 380-386, (2013).

[12] T. Hiroyasu, M. Miki, M. Negami, “Parallel distributed genetic algorithm with randomized migration rate” (Japanese), The Science and Engineering Review of Doshisha University, 40(2), pp. 25-34, (1999).

[13] L. J. Eshelman, J. D. Schaffer, “Real-coded genetic algorithms and interval-schemata”, Foundations of Genetic Algorithms 2, Morgan Kaufman Publishers, pp. 187-202, (1993).

[14] A. H. Wright, “Genetic algorithms for real parameter optimization”, Foundations of Genetic Algorithms, Morgan Kaufman Publishers, pp. 205-218, (1991).