Fundamentals for Real World Applications of Metaheuristics:
The vehicular case
Ministerio de Economía y Competitividad (Spain)
TIN2011-28194
http://roadme.lcc.uma.es

Málaga, June 2013

Executive Summary

TITLE: **D3.1.2: GPU-enabled design strategies and implementations of metaheuristics**

PAPERS RELATED:

- Pedemonte, M., Luna, F., and Alba, E. (2014). **Systolic genetic search, a systolic computing-based metaheuristic.** Soft Computing, 1-23.
- Pedemonte, M., Luna, F., and Alba, E. (2013). **New Ideas in Parallel Metaheuristics on GPU: Systolic Genetic Search.** In Massively Parallel Evolutionary Computation on GPGPUs (pp. 203-225). Springer Berlin Heidelberg.
- Pedemonte, M., Luna, F., and Alba, E. (2014). **Systolic Genetic Search for Software Engineering: The Test Suite Minimization Case.** In Applications of Evolutionary Computation (pp. 678-689). Springer Berlin Heidelberg.

ABSTRACT: This deliverable presents an in-depth study of a novel parallel optimization algorithm specially designed to run on Graphic Processing Units (GPUs). The underlying operation relates to systolic computing and is inspired by the systolic contraction of the heart that makes possible blood circulation. The algorithm, called Systolic Genetic Search (SGS), is based on the synchronous circulation of solutions through a grid of processing units and tries to profit from the parallel architecture of GPUs to achieve high time performance. SGS has shown not only to numerically outperform a random search and two genetic algorithms for solving the Knapsack Problem over a set of increasingly sized instances, but also its parallel implementation can obtain a runtime reduction that, depending on the GPU technology used, can reach more than 100 times. A study of the performance of the parallel implementation of SGS on four different GPUs has been conducted to show the impact of the Nvidia's GPU compute capabilities on the runtimes of the algorithm.

GOALS:

1. Design a novel parallel technique to run on GPUs.
2. Study the features of this new algorithm.
3. Test its behaviour on benchmark problems and real-world ones (such as software engineering problems).

CONCLUSIONS:

1. SGS variants designed are highly effective for an extensive number of instances of benchmark problems.
2. The parallel implementation on GPU of the proposed algorithm has achieved a high performance.
3. The experimental evaluation on seven real-world programs shows that SGS is highly effective for the Test Suite Minimization Problem, as it obtains the optimal solution in almost every single run for all the tested software.

RELATION WITH PAST DELIVERABLES:

PRE: D3.3.1 (advisable reading)