Designing Embedded Parallel Systems with Parallel Genetic Algorithms

E-G.TALBI & T. MUNTEAN
IMAG - *Laboratoire de Génie Informatique* (*)
University of Grenoble

Genetic algorithms are general purpose optimisation and search techniques inspired from biological principles of evolution of natural systems, and introduced by Holland twenty years ago. Genetic algorithms are good candidates of efficient heuristics that behave through complex spaces in a near optimal way. Standard genetic algorithms with large populations suffer from lack of efficiency. Development of massively parallel architectures made them very popular in the very last years. They have recently been applied to combinatorial optimisation problems in various fields, such as, for instance, the travelling salesman problem, the optimisation of connections and connectivity of neural networks and reconfigurable parallel machines, and classifier systems. We have proposed a parallel algorithm to speed up the genetic process and prove that in various fields of applications optimisation problems may be solved quite efficiently.

The goal of the work presented is to build efficient parallel genetic algorithms for embedded parallel systems. The parallel genetic algorithm proposed is generic, that is, it can be used for any optimisation problem. The real time path planning problem for mobiles is taken as an example of application.

We deal with such a (non trivial) example because designing a real time path planner is a classical exercise in robotics research and it remains a very active field in robotics. There are two main ways to deal with this problem: the global and the local approaches. The global approaches suppose that a complete representation of the configuration space has been computed before looking for a path. The global approaches are complete in the sense that if a path exists it will be found. Unfortunately, computing the complete configuration space is very time consuming, worst, the complexity of this task grows exponentially as the number of degrees of freedom increases.

The local approaches need only partial knowledge of the configuration space. The decisions to move the robot are taken using local criteria and heuristics to chose the most promising directions. Consequently, the local methods are much faster. Unfortunately, they are not complete, it may happen that a solution exists and is not found. The local approaches consider planning as an optimisation problem, where constructing a path to the target corresponds to the optimisation of some given function. As any optimisation technique, the local approaches are subject to get trap in some local optima, where a path to the goal has not been found and from which it is impossible or, at least, very difficult to escape.

---

* Authors adress: IMAG-LGI, BP53X -F38041 GRENOBLE; e-mail: muntean@imag.fr

Today most of the robot motion planners are used off-line: the planner is invoked with a model of the environment, it produces a path which is passed to the robot controller which in turn execute it. In general, the time necessary to achieve this loop is not short enough to allow the robot to move in a dynamic environment (moving obstacles). Our goal is to try to reduce this time in order to be able to deal with real time path planing in dynamic environments.

In order to do so, we use a method, called "ARIADNE'S CLEW algorithm", to build a global path planner based on the combination of two parallel genetic algorithms: an EXPLORE algorithm and a SEARCH algorithm. The purpose of the EXPLORE algorithm is to collect information about the environment with an increasingly fine resolution by placing landmarks in the searched space. The goal of the SEARCH algorithm is to opportunistically check if the target can be reached from any given placed landmark.

The ultimate goal of a planner is to find a path in the configuration space from the initial position to the target. However, while searching for this path, an interesting sub-goal to consider may be to try to collect information about the free space and about the possible paths to go about that space. The ARIADNE'S CLEW algorithm try to do both at the same time. An EXPLORE algorithm collects information about the free space with an increasingly fine resolution, while, in parallel, a SEARCH algorithm opportunistically check if the target can be reached. The EXPLORE algorithm works by placing landmarks in the searched space in such a way that a path from the initial position to any landmark is known. In order to learn as much as possible about the free space the EXPLORE algorithm try to spread the landmarks all over the space. To do so, it tries to put the landmarks as far as possible from one another. For each new landmark produced by the EXPLORE algorithm the SEARCH algorithm checks with a local method if the target may be reached from that landmark. The ARIADNE'S CLEW algorithm is very fast, however, we will show that it is a complete planner which will find a path if one exits. The resolution at which the space is scanned and the time spend to do so, automatically adapts to the difficulty of the problem.

The ARIADNE'S CLEW algorithm is shown to be very fast in most cases allowing planning in dynamic environments. Hence, it is shown to be complete, which means that it is sure to find a path when one exists. Three levels of parallelism may be found:

- a top level of parallelism, SEARCH and EXPLORE may run in parallel,
- both SEARCH and EXPLORE need to run a genetic algorithm,
- each generation may calculate in parallel the fitness function.

We have implemented our parallel genetic algorithm the on a massively parallel distributed memory machine, based on transputers. Using 128 processors (T800 transputers), the mean planning time was under the second. We have a compromise between the number of processors used and the efficiency of the algorithm. Work is ongoing on the automatic derivation of the machine size (number of processors and their type) in function of the response time required by such applications in the field of real-time embedded system design.

This example will be used at the workshop to illustrate the methodology behind the design of some classes of parallel real time embedded systems using parallel genetic algorithms.

[MT 91] T.Muntean, E-G.Talbi **"A parallel genetic algorithm for process-processors mapping"**, *Int. Conf. on High Speed Computing II, M.Durand and F.El Dabaghi (Editors), Elsevier Science Pub., North-Holland, Oct 1991.*

[ABMT 92] J.M.Ahuactzin, P.Bessière, E.Mazer, E-G.Talbi **"Parallel robot motion planning in a dynamic environment"**, *International Conference on Parallel and Vector Processing CONPAR92-VAPP V, Lyon, France, Sep 1992.*