A DISTRIBUTED GENETIC ALGORITHM TO OPTIMIZE POLYGON PACKING

K.M. LIYANAGE AND A. RATNAWEERA

Department of Electrical and Electronic Engineering, Faculty of Engineering, University of Peradeniya

In many industrial applications, it is necessary to generate parts by cutting them from stock material. This involves generating non-overlapping layouts of shapes, which are in general irregular and non-convex. Although many attempts have been made to automate this task, human operators are still superior. On the other hand, programs that improve human generated layouts are available and are used in production environments. In this paper, we are using a distributed genetic algorithm to generate dense polygon layouts, which can exploit massive computing power available in parallel computing architectures. Also, in contrast to local optimization approaches, the genetic algorithm traverse the full search space, and can even be used to provide local optimization programs with starting solutions. In fact, this is what human operators do in many circumstances.

Theoretically, the problem in hand is known as the *polygon packing problem*, or sometimes as the *containment problem*. A related problem is overlap minimization, where a measure of overlap is minimized under a given set of constraints. Although we consider only polygons, circular and other curved shapes can also be approximated to any required degree of accuracy using line segments. Attempts have been made to generate better polygon layouts using simulated annealing, boundary matching, database driven layouts, hybrid approach, and serial genetic algorithms. However, our approach is different from those, because we consider parallel implementation, and our handling of selection, crossover and mutation is different.

In apparel industry, the orientation of shapes is also important; therefore, constraints are imposed on rotations. This is different from similar problems involving metal glass, or leather, where arbitrary orientations of shapes are allowed.

In the present context, we focus our attention on optimizing the length L of the containing rectangle with fixed width, by changing the set of transformations applied to each polygon. A transformation is defined by a rotation and a translation. If N is the number of solutions, the search space is R^{3N}, and the problem is NP hard, even for small values of N. We adopt a parent-child model in the implementation, where the parent process manages the tasks done by each child process, but doesn't get involved in the bulk of the computation. Computationally expensive tasks are handled by child processes. This avoids the parent task becoming a bottleneck.

The key feature of our genetic algorithm is that it is embarrassingly parallel. It is designed to work even when execution times of various threads are different. Also, it is inherently fault tolerant, and therefore stopping a process doesn't have any effect of its execution. After generating a random population, and after a crossover or mutation step, some solutions contain overlaps. We considered various methods of resolving, and studied the performance of the algorithm with each problem. Performance of the algorithm was measured with problems of different scales and shapes, and under varying degrees of parallelism. Also, the effect of different selection, crossover and mutation methods are also considered.