MPBO - A Distributed PBO Solver

Nuno Santos, nunocsantos@ist.utl.pt
Technical University of Lisbon

***Abstract*—Parallel computing has been the subject of intensive research in the last decade and is increasingly being chosen as a solution for developing applications that require high computational power, such as the *Boolean Satisfiability* (SAT) and *Pseudo-Boolean Optimization* (PBO) problems. Research in SAT solvers has obtained relevant results in the last years, achieving significant reductions in execution times. Unfortunately, hard instances of SAT require large computational power and even efficient SAT solvers take huge execution times to obtain their solution. Therefore, SAT solvers adaptation to parallel computing systems began to be the subject of considerable research and there already exist several distributed versions of popular SAT solvers. However, the absence of distributed solvers for the PBO problem is notorious. This work intends to contribute and encourage the research into distributed solutions to solve the PBO problem. The goal of this work is to propose a distributed Pseudo-Boolean Optimization Solver, named MPBO solver, based on MPI (*Message Passing Interface*) and focused on an efficient search space partition, more specifically the partition of the problem optimization search space. The proposed solver achieved significant reductions in the time to solve hard PBO instances, when compared to the MiniSat+ and pwbo solvers.**

I. Introduction

The well known *Boolean Satisfiability* (SAT) and *Pseudo-Boolean Optimization* (PBO) problems gained significant attention in the last years, due to their possible application in many domains, such as software and hardware verification. Since then, several algorithmic solutions have been proposed to solve both problems and many of them proved to be very efficient when solving several instances of the problems. Many SAT algorithmic solutions (known as SAT solvers), for instance `GRASP` [1] and `Chaff` [2], contributed with several techniques to improve the SAT resolution. Due to the relevant results on SAT research, PBO researchers embraced the SAT solvers efficiency and proposed efficient ways to solve PBO instances by extending SAT solvers to handle them.

Although SAT solvers are becoming more sophisticated to reduce their execution time, adopting improvement techniques such as *clause learning*, *adaptive branching*, *restarts* and *non-chronologically backtracks*, the demand for more computational power led SAT researchers to explore solutions taking advantage of parallel computing systems. Distributed systems, such as *clusters* and *grids*, are a popular type of parallel computing environments and are target of great adoption in the parallel developers community, since they allow the use of several remote resources, connected through a network, and enable the execution of distributed algorithms that can divide a problem for a concurrent resolution in such resources. Many popular SAT solvers have been migrated to such environments, adopting the well known *Task Farm* approach, which proved to be a successful solution to partition SAT problem instances for concurrent searches. However, unlike SAT solvers, the migration of PBO solvers to distributed systems has not been well explored yet. Taking such gap in consideration, this work is intended to give one more step for such migration by proposing the MPBO (*Message Passing Pseudo-Boolean Optimization*) solver, an efficient distributed PBO solver that obtains significantly reduced execution times when solving hard problem instances. The MPBO solver was implemented resorting to the industry's defacto MPI (*Message Passing Interface*) API to allow its portability through the different distributed environments and is based on the core of the well known `MiniSat+` [3] solver. The solver presented achieved speedup values of 500 in the time to solve hard PBO instances, when compared to `MiniSat+`, and speedups of 6 when compared to the parallel `pwbo` solver [4].

From the analysis of the objective function of a PBO problem, it is possible to calculate an interval where all solution candidates for the problem are included, known as the *optimization interval*, and such interval represents the optimization search space that must be explored. The approach behind the MPBO solver engine is the partition of the optimization search space by the available resources in a distributed environment, to be explored concurrently. Based on the well known Task Farm approach, the solver is composed by two entities. The workers, which are responsible to computed the assigned tasks, and the master which manages the workers through the computation of the problem. The workers are composed by a mechanism to communicate with the master, to receive tasks and send the calculated results, and the modified core of the `MiniSat+` solver to compute the tasks received. The master is responsible to assign the necessary tasks and contains a mechanism to identify the tasks to compute to reach the optimal solution for the problem. Each task corresponds to a specific restriction to the objective function of the problem that must be computed, representing a specific value of the optimization interval, and each task results in a *sat* or *unsat* response, according to the satisfiability of the given restricted problem. Through the problem optimization search, the master manages the optimization interval according to the results obtained by the computed tasks and restricts such interval until only one value remains, which is considered the optimal solution for the problem. The MPBO solver features two task assignment approaches: the simple approach, where the optimization search space is divided in a binary search fashion, and the optimized approach, where the optimization search space is divided considering the objective function to minimize. The optimized approach reduces significantly the number of task assignments performed during the execution of the solver and achieves an average speedup gain of almost 80% than the simple assignment approach.