Solving Multi-objective Pseudo-Boolean Problems

Martin Lukasiewycz, Michael Gläß, Christian Haubelt, and Jürgen Teich

Hardware-Software-Co-Design
Department of Computer Science 12
University of Erlangen-Nuremberg, Germany
{martin.lukasiewycz, glass, haubelt, teich}@cs.fau.de

Abstract. Integer Linear Programs are widely used in areas such as routing problems, scheduling analysis and optimization, logic synthesis, and partitioning problems. As many of these problems have a Boolean nature, i.e., the variables are restricted to 0 and 1, so called Pseudo-Boolean solvers have been proposed. They are mostly based on SAT solvers which took continuous improvements over the past years. However, Pseudo-Boolean solvers are only able to optimize a single linear function while fulfilling several constraints. Unfortunately many real-world optimization problems have multiple objective functions which are often conflicting and have to be optimized simultaneously, resulting in general in a set of optimal solutions. As a consequence, a single-objective Pseudo-Boolean solver will not be able to find this set of optimal solutions. As a remedy, we propose three different algorithms for solving multi-objective Pseudo-Boolean problems. Our experimental results will show the applicability of these algorithms on the basis of several test cases.

1 Introduction

Solving 0-1 Integer Linear Programs (0-1 ILP) came to the field of vision over the past years. This problem class is a special case of Integer Linear Programs (ILP) and is also termed as Pseudo-Boolean (PB) \cite{1}. In particular a Pseudo-Boolean problem is an optimization problem with a linear objective function and a set of linear constraints in which the coefficients are integers and the variables are restricted to 0 and 1. Despite the restriction of the variables to Boolean values the expressiveness is equal to ILPs which can be formulated as Pseudo-Boolean problems by using a binary encoding.

The Boolean nature of Pseudo-Boolean problems is connecting these strongly to the Satisfiability problem (SAT) in conjunctive normal form \cite{2}. The Satisfiability problem can easily be converted to a Pseudo-Boolean problem with an empty objective function in which for each clause a greater-zero constraint is added. The 0-1 Integer Linear Programming is, in fact, one of Karp’s 21 NP-complete problems \cite{3}. On the other hand, converting efficiently PB constraints into clauses is a non-trivial problem that can result in an exponential number of clauses.

There are several PB solvers that are borrowing techniques from state-of-the-art SAT solvers which became essential in the field of Electronic Design Automation \cite{4}. These specialized PB solvers are based on the DPLL backtracking algorithm \cite{5} and benefit

\footnote{Supported in part by the German Science Foundation (DFG), SFB 694.}
from the improvements on the field of SAT-solving of the recent years like the non-chronological backtracking [6], watched literals [7], or an efficient conflict learning scheme [8]. As a matter of fact it is validated that specialized PB solvers are superior to generic ILPs, mostly if the underlying problem has a Boolean nature [9].

PB solvers have their applications among many real-world applications like routing problems, scheduling analysis and optimization, logic and system level synthesis, and partitioning problems. Some of these applications like the problem of system level synthesis [10] can contain more than one objective function, e.g., if the system is optimized by its power consumption, area usage, and the monetary costs. In the case of multi-objective optimization the goal is not to find optimal solutions corresponding to each objective function, but to find the set of optimal solutions the so called Pareto-optimal solutions. A solution is called Pareto-optimal if there exists no other solution that is better or equal in all objectives and at least better in one objective, i.e., no other solution dominates the Pareto-optimal one. As the search space in Pseudo-Boolean problems is finite the number of Pareto-optimal solutions is also finite. Figure 1(a) illustrates the Pareto-optimal solutions of a problem with two objective functions. A PB solver optimizes at most one objective function and will not find these Pareto-optimal solutions as preference-based approaches do not find the trade-off solutions. In the case of system level synthesis a designer is interested in the full set of Pareto-optimal solutions containing the trade-off solutions to make an appropriate choice for one implementation.

This paper is dedicated to the multi-objective Pseudo-Boolean problem in which we propose three different algorithms for solving multi-objective Pseudo-Boolean problems and compare them on the basis of several test cases. The first algorithm is an iterative search with a common PB solver by restricting the search space by upper bounds. The second algorithm extends a DPLL backtracking algorithm such that it sifts through the valid search space and at the same time prunes evidently not optimal solutions. The third algorithm is using a translation into the Satisfiability problem such that a common SAT solver finds one solution that fulfills the constraints. To ensure a convergence to the Pareto-optimal solutions, the found and dominated solutions are excluded from the ongoing search by appending additional clauses.

The rest of the paper is organized as follows: Section 2 gives a short introduction to the functionality of modern PB solvers, and Section 3 will formally state the problem this paper is dedicated to. In Section 4 the three algorithms for solving multi-objective