

## *Dissertation Proposal*

# **Decision Diagrams for Optimization**

Andre Augusto Cire

Wednesday, November 6, 2013

10:30 am

324 GSIA

Decision diagrams are compact graphical representations of Boolean functions originally introduced for applications in circuit design, simulation, and formal verification. Recently, they have been considered for a variety of purposes in optimization and operations research. These include facet enumeration in integer programming, maximum flow computation in large-scale networks, solution counting in combinatorics, and learning in genetic programming techniques.

In this thesis we develop new methodologies based on decision diagrams to tackle *discrete optimization problems*. A decision diagram is viewed here as a graphical representation of the feasible solution set of a discrete problem. Since such diagrams may grow exponentially large in cases of most interest, we work with the concept of *approximate decision diagrams*, first introduced by Andersen et al (2007). An approximate decision diagram is a graph of parameterized size that represents instead an over-approximation or under-approximation of the feasible solution set. Thus, it can be used to obtain either bounds on the optimal solution value or primal solutions to the problem.

As our first contribution, we provide a modeling framework based on dynamic programming that can be used to specify how to build a decision diagram of a discrete optimization problem and how to approximate it, which facilitates the encoding process of a problem to a diagram representation. We then present a branching scheme that exploits the recursive structure of an approximate diagram, establishing a novel generic solver for discrete optimization problems. We analyze the performance of our solver in classical discrete optimization problems, identifying the types of structures that are favorable to our method.

In our second contribution, we focus on the application of approximate diagrams to particular domains; namely, to *sequencing problems*, common in the context of routing and scheduling, and to *timetable problems*. We show that, besides the computation of bounds, approximate decision diagram can be used to deduce non-trivial constraints of a problem, such as precedence relations between jobs in scheduling applications. We show that such inference can be incorporated into state-of-the-art solvers and speed-up the optimization process by orders of magnitude.

Finally, we propose new parallelization strategies that exploits the recursive structure of an approximate diagram. These strategies decouple a problem in a naturally loose fashion and allow for more effective load balancing heuristics when considering hundreds of computer cores.