(DRAFT) Implicit enumeration with dual bounds

Nelson Frew

Introduction

The field of discrete optimisation (CITE) studies problems that model discrete decision making. Mixed-Intger Programming refers to a set of modelling techniques for discrete problems (CITE), where one aims to maximise/minimise an objective function $z = cx$, where c is our problem data, and x is the set of variables we can change to affect z's value. Generally, constraints on how we can optimise the objective function are linear, and variables can be specified to be integer or continuous.

Optimisation problems with linear-constraints with continuous variables are known as Linear Programs (LPs), which are solvable in polynomial time (CITE). Mixed-Integer Programs (MIPs) are characterised by having one more more variables constrained to be integer, resulting in an NP-Hard problem.

The state-of-the-art method for solving mixed-integer programs is the Branch and Bound approach, which uses LP relaxations to achieve *implicit enumeration* of the search space. Finding the LP relaxation of a MIP simply involves removing the integer constraints on our variables, and solving the resulting LP. Because the feasible region of continuous solutions naturally subsumes the set of feasible solutions for integers, the LP relaxation optimal value will be, for a maximisation problem, *at least* the value of the optimal value for the original MIP. This provides an upper bound in our case, which we refer to as the *dual bound.* Given our solved relaxtion, we contrain variables to "tighten" the continuous feasible region closer to the maximum integer solution. By solving LP relaxation again for this constrained subproblem, we can see if an optimal integer solution can exist in with these constraints: if our dual bound is lower than our best known value so far, we can *prune* this subproblem and switch to another constrained subproblem. Otherwise, if this subproblem has a dual bound above our best known value, we continue constraining and solving the relaxation until a new feasible solution is found, or a further constrained subproblem of this is pruned.

While the Branch and Bound is at worst case exponential-time, there exists Approximation Algorithms (AAs) which can provide feasible in polynomial-time. Further, AAs provide a guarantee on the quality of the solution. An α-approximation is an AA that guarantees that the solution value will always be within a constant factor α of the optimal solution. As a result, for a maximisation problem, we can analytically derive an upper bound on the optimal value from the lower bound guarantee on the optimal solution. In this way, we can use AAs in lieu of a LP relaxation to obtain valid dual bounds on our optimal solution. Because we are leveraging guarantees on the optimal value, the dual bounds we obtain are also guaranteed to be within a level of accuracy, relative to the optimal value. This is distinct from LP relaxations, which can provide arbitrarily poor dual bounds.

In this project, we investigate methods of effectively using dual bounds provided by AAs for the 0,1 Knapsack. To do this, we investigate known approximation schemes, and methods to construct high quality dual bounds from them, as well as devising branching strategies within its Branch and Bound.