Feasibility and Improvement in Mixed Integer Programming: A Case Study

Marcus Poggi de Aragão (PUC-Rio)

Eduardo Uchoa (UFF)

Fernanda Menezes (GAPSO Inc.)

May, 2006.

We consider the problem of finding high quality feasible solutions for mixed integer programs (MIPs) of the form $\min\{cx : A_i x \geq b_i,\ i \in M,\ x_j \text{ integer } \forall j \in J_I\}$. Our approach follows the ideas in *Local Branching* and *VNS* to obtain local improvements. This settles as first challenge finding a feasible solution. We propose to overcome this difficulty by gradually adding the MIP constraints and by introducing penalties for not satisfying them.

Let MIP_{curr} be the MIP above considering only subset M_{curr} of constraints. Let also s_i with cost p_i be slack variables added to the MIP. These slack variables are only present for $i \notin M_{curr}$. Our algorithm starts by setting M_{curr} to M_0 and finding a feasible solution for MIP_{curr}. Here M_0 may be the empty set. It iterates by sequentially adding a constraint $i \notin M_{curr}$ performing local improvements until s_i is zero, i.e. a feasible solution for MIP_{curr} was found. We set M_{curr} to $M_{curr} \bigcup \{i\}$ and a second phase of local improvements thens starts from the partially feasible solution just found. The iterations are repeated until M_{curr} becomes M.

Several questions on how this approach should perform best are then raised. Should we try to arrive to a feasible solution to MIP as fast as possible and then spend the time available improving on globally feasible solutions ? How many constraints should be added at each iteration ?

We try to answer these questions on a case study: the planning of schedules of helicopter flights for serving the platforms of the Brazilian oil company Petrobras. The suitability of this problem to this experience comes from the following characteristics. The associated MIP formulation used has a reasonable integrality gap (about 20%), it is highly symmetric, has continuous, binary and integer variables, easy and hard constraints and most of all, in all parametrizations we tried, CPLEX was unable to find a feasible solution after running for more than 24 hours.

The problem aims at finding the time-table of the flights for a week. This time-table is to be used for a whole year and it must satisfy several types constraints including frequency and area congestion. The objective function may emphasize minimizing the required fleet, quality of the transportation service, and/or security measures.

We present an extensive computational experience for different sets of scenarios defined by forecasts for platform production, location and technology use, and also forecasts for helicopters' types and prices for hiring. Feasible solutions could be found in less than one hour, although this strategy showed not to be the best way to reach the best solutions. The resulting code uses CPLEX 10.0 as MIP solver and is able to produce high quality solutions appreciated by the end user in such a way that investing in long term research on this problem is now a priority.