Analyzing Infeasible MIPs

CORAL Seminar Series

Menal Guzelsoy

Feb 17, 2005

References

- John W. Chinneck, *Analyzing Infeasible Optimization Models*, A tutorial for CORS/INFORMS, 2004.
- O. Guieu and J.W. Chinneck, *Analyzing Infeasible Mixed-Integer and Integer Linear Programs*, INFORMS Journal on Computing, vol. 11, no. 1, pp. 63-77, 1999.
- J.W. Chinneck, *Finding a Useful Subset of Constraints for Analysis in an Infeasible Linear Program*, INFORMS Journal on Computing , vol. 9, no. 2, 1997.

Infeasibility Isolation

- Why is it *infeasible*?
  - Applications
    - Training neural networks
    - Radiation therapy dose planning
    - Design/Analysis of protein folding potentials
    - Statistics
    - Approximating NP-hard problems
    - Automatic test assembly
    - Backtracking in constraint logic programming
  - The answer may not be trivial in large models

Infeasibility Isolation

- However, there are algorithms available to isolate the infeasibility
  - to a subset of the constraints(row and column bounds, integer restrictions) (IIS)
  - to help diagnose the cause and repair the model
- IIS: Irreducible Inconsistent System
  - infeasible but any subset is feasible
  - should not contain any constraints which do not contribute to the infesibility.
  - there may be more than one
- These algorithms require to solve a set of slightly modified MILPs.
  - SYMPHONY's warm-starting capability.

Infeasibility Analysis for LPs

- Two algrotihms that guarantee the isolation of an IIS:
  - Deletion Filter
  - Additive Method

Deletion Filter

```
Input: an infeasible set of constraints
For each constraint in the set
–drop the constraint from the set
–test the feasiblity of the reduced set
—-If feasible Then return dropped constraint to the set
—-Else drop the constraint permanently
Output constraints of a single IIS.
```

Additive Method

C: ordered set of constraints in the infeasible model
T: the current test set of constraints
I: the set of IIS members identified so far

```
Input: an infeasible set of constraints C
Step 0: Set T = I = $\emptyset$
Step 1: Set T = I
–For each constraint $c_i$ in C
—Set T = T $\cup$ $c_i$
—If T infeasible Then
—-Set I = I $\cup$ $c_i$
—-Go to Step 2
–End For
Step 2:If I feasible Then go to Step 1
Output: I is an IIS.
```

There are some other algorithms to speed the isolation but have to be used with the basic methods.

- grouping constraints
- elasticizing constraints and using elastic filter
- bound-tightening
- sensitivity filter
- interior point methods
- reciprocal filter, simplex pivoting.

Infeasibility Analysis for MIPs

- complicated due to the presence of the integer restrictions
- the deletion filter and the additive method are very effective for LP's because
  - LP solvers are able to decide the feasibility status of a model accurately
  - the procedure is fast due to advanced starts/basis re-use
- but for MIPs,
  - time and memory limit/branch and bound tree may grow infinitely hence no guarantee about the feasibility status of a subproblem
  - current softwares have to solve each subproblem from scratch
  - There are three subsets of the constraints:
    - LC: linear row constraints
    - BD: variable bounds
    - IR: integer restrictions

Deletion Filtering for MIPs

main modifications to the algorithm:

- if a subproblem is abandoned due to time or memory limit, then,
  - common assumption is that the subproblem is feasible
  - no guarantee of identifying an IIS, instead, will get an IS: Infeasible Subsytem
  - however, IS is still useful since it will limit the effort to a smaller portion of the entire system.
  - label the corresponding constraint as *dubious* and move it to the output set in order to satisfy an IS.
- the speed is affected by the order of the constraints:
  - if all BDs are not both upper and lower bounded then test the BDs at the end. so that they will remain in system to limit the branching nodes while testing LC and IR.
  - IRs may be tested before LCs hoping that some of the IRs will be eliminated early.

Additive Method for MIPs

main modifications to the algorithm:

- assuming initial LP is feasible, begin with $\mathrm{T} = \mathrm{LC} \cup \mathrm{BD}$
- may not directly identify the *dubious* constraints, because the test set is maintained in a feasible or indeterminate state
- then, the output is an IS (may also be an IIS, not known)
- Dynamic reordering variant:
  - all the intermediate subproblems are feasible until infeasiblity is obtained
  - when a subproblem is feasible, scan all constraints past the current constraint, and add all constraints satisfied at current solution.
  - this may reduce the number of feasible subproblems solved

Additive/Deletion method for MIPs

- both methods can be combined
- use additive method until the infeasiblity is reached
- then switch to deletion filter
- this will also identify the *dubious* constraints

Speed-ups for MIPs

- Preprocessing of MILP using information from the initial branch and bound tree:
  - no IIS has IR set identical to the set of IRs satisfied at any intermediate node
  - mark sensitive (having nonzero duals/reduced costs at Phase I) LCs and BDs at all leaf nodes, then, IR $\cup$ {marked LCs}$\cup${marked BDs} is infeasible
  - LC $\cup$ BD $\cup$ IRs on all branching variables is infeasible
- Grouping constraints
  - for instance: during deletion filtering, constraints can be dropped in groups.
- Safety Bounds
  - extra BDs to prevent nontermination. if it is infeasible, then the output is IS.