Sparse Simplex and LPs with Embedded Network Structure

CORAL Seminar Series

Scott DeNegre

29 June 2005

Reference

Gülpinar, N., G. Mitra, and I. Maros 2002. Creating Advanced Bases For Large Scale Linear Programs Exploiting Embedded Network Structure. *Computational Optimization and Applications*, **21**(1), 71-93.

Introduction

Definition 1. *An* embedded network *is a structure appearing within a problem consisting of a subset of constraints and variables which define a network flow problem. The remaining constraints are referred to as* side constraints*.*

- The main goal of the paper is to exploit the embedded network structure to create an advanced basis via decomposition methods.

Problem Statement

Consider the primal LP

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & Ax = b \\
& l \le x \le u
\end{aligned}
\tag{1}
$$

where $A \in \mathbb{R}^{m \times n}$, $c, x, l, u \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$.

Network Decompostion

If we assume that we can detect a submatrix of network rows and columns of A is detected, and let N denote thh pure network structure, we can rewrite (1) as

$$\begin{aligned} \min \quad & z_0 = {c'}^T x' + {c''}^T x'' \\ \text{s.t.} \quad & N x' = b' \\ & S x' + T x'' = b'' \\ & l' \le x' \le u' \\ & l'' \le x'' \le u'' \end{aligned} \tag{2}$$

where

$$\begin{aligned} m &= m_1 + m_2, n = n_1 + n_2, \\ N &= [n_{ij}] \in \mathbb{R}^{m_1 \times n_1}, S = [s_{ij}] \in \mathbb{R}^{m_2 \times n_1}, T = [t_{ij}] \in \mathbb{R}^{m_2 \times n_2}, \\ c', \quad x', \quad & l', u' \in \mathbb{R}^{n_1}, c'', x'', l'', u'' \in \mathbb{R}^{n_2}, b' \in \mathbb{R}^{m_1}, b'' \in \mathbb{R}^{m_2} \end{aligned}$$

Solving LPEN with Advanced Basis

Two things we know:

- For large-scale LPs, using an advanced basis improves sparse simplex performance
- Calculation of the initial basis is *very* important

$\Rightarrow$ Construct an algorithm which exploits the embedded network structure to create an advanced basis!

Decomposition Methods

Lagrangean Relaxation

- Creates a pure network flow model by adding the non-network constraints into the objective function with Lagrangean penalties
- A series of minimum cost NFPs corresponding to different values of Lagrangean multipliers are solved iteratively

Benders Decomposition

- Decomposes the LP problem in to a master and a subproblem
- Cuts obtained from the subproblem are added to the master problem at each iteration

The Algorithm

- *Step 1: Preprocessing and Scaling*
- *Step 2: Network Detection*
  - Detect the network structure and decompose the problem as in 2
- *Step 3: Solve the Decomposition Problem and Create a Starting Basis*
  - Apply Lagrangean Relaxation **OR** Benders Decomposition
- *Step 4: Complete Sparse Simplex Solution*
  - Process the LPEN applying simplex method of choice using starting basis obtained in Step 3

Lagrangean Relaxation

Relaxing the side constraints $Sx' + Tx'' = b''$ yields the Lagrangean relaxation

$$\begin{aligned} \min \quad & z_{L(\lambda)} = \lambda^T b'' + \left(c'^T - \lambda^T S\right) x' + \left(c''^T - \lambda^T T\right) x'' \\ \text{s.t.} \quad & Nx' = b' \\ & l' \leq x' \leq u' \\ & l'' \leq x'' \leq u'' \\ & \lambda \in \mathbb{R}^{m_2} \text{ and unrestricted} \end{aligned} \tag{3}$$

Note that (3) is a pure NFP and can be solved efficiently using network simplex.

Lagrangean Relaxation

The dual of the the original LP (2) is given by

$$\begin{aligned} \max \quad & b'^T\pi + b''^T\lambda + l'^T\sigma_1 + l''^T\sigma_2 - u'^T\eta_1 - l''^T\eta_2 \\ \text{s.t.} \quad & N^T\pi + S^T\lambda + \sigma_1 - \eta_1 = c' \\ & T^T\lambda + \sigma_2 - \eta_2 = c'' \\ & \sigma_1, \sigma_2, \eta_1, \eta_2 \geq 0 \end{aligned} \tag{4}$$

- This problem is the same as the dual of (3), with different RHS values
- The Lagrangean dual problem of (2) w.r.t. the side constraints is to find multipliers λ^* that maximizes $z_{L(\lambda)}$

$$z^*_{L(\lambda^*)} = \max_{\lambda} \left\{ \min_{(x',x'')} \left(c'^T - \lambda^T S\right) x' + \left(c''^T - \lambda^T T\right) x'' \right\} \tag{5}$$

Finding the Multipliers

To determine the Lagrangean multipliers λ, we solve the LP

$$\begin{aligned} z^*_{L(\lambda^*)} &= \max w \\ \text{s.t.} \quad & w \le f_j + \lambda^T g^j, \quad j = 1, \ldots, K \qquad (6) \\ & \qquad (7) \end{aligned}$$

where f_j is the objective value of (2), g^j is the subgradient of the jth basic solution

$$f_j = {c'}^T {x'}^j + {c''}^T {x''}^j, \; g^j = b'' - S{x'}^j - T{x''}^j$$

and K is the number of all basic solutions of the Lagrangean problem.
Note:

- $z^*_{L(\lambda)} \le z^*_0$ for all λ
- $\exists$ Lagrangean multipliers λ^* such that $z^*_{L(\lambda^*)} = z^*_0$ (not true in general)

Crash Procedures

- Crash procedures are designed to create an initial basis to provide an advanced starting point
- It is well known that a starting basis with multiple structural (as opposed to logical) variables needs fewer iterations an less time to find an optimal solution
- *Triangular* crash procedures find a basis matrix which has as many structural variables as possible in such a way that the resulting basis matrix has a triangular form with a zero-free diagonal
  - The triangularity of the basis matrix ensures that a factored inverse representation of the basis with a minimum number of non-zeros can be trivially created

Network Based Crash Procedures

- For given λ the Lagrangean relaxation is a minimum cost NFP $\Rightarrow$ feasible or optimal basis has triangular form
  - Leads to a lower triangular crash procedure considering only variables which are basic in the network optimal solution
- Construct initial basis from basic variables of the network problem and logical variables of non-network rows, apply network based crash procedure (i.e. CNET2)

Benders Motivation

The embedded NFP can be considered as a 2-stage problem in which the network and non-networks part are the 1st and 2nd stages, respectively.

$\Rightarrow$ Use Benders decomposition to create an advanced basis for solving the original LPEN problem

Benders Decomposition

We can split the original problem (2) into a master problem

$$
\begin{aligned}
\min \quad & z_M = {c'}^T x' \\
\text{s.t.} \quad & N x' = b' \\
& l' \le x' \le u'
\end{aligned}
\tag{8}
$$

and a subproblem

$$
\begin{aligned}
\min \quad & z_S = {c''}^T x'' \\
\text{s.t.} \quad & T x'' = b'' - S {x'}^* \\
& l'' \le x'' \le u''
\end{aligned}
\tag{9}
$$

where ${x'}^*$ is an optimal solution of (8).

Subproblem Dual

The dual of the subproblem (9) is given by

$$\begin{aligned} \max \quad & \pi_1^T(b'' - Sx'^{*}) - \pi_2^T u'' + \pi_3^T l'' \\ \text{s.t.} \quad & \pi_1^T T \le c'' \\ & \pi_1 \text{ free} \\ & \pi_2, \pi_3 \ge 0 \end{aligned} \tag{10}$$

The Cuts

- From duality theory, we have a feasibility cut of the form

$$\pi_1^T(b'' - Sx'^*) - \pi_2^T u'' + \pi_3^T l'' \leq 0$$

- If we let θ denote the smallest value of the upper bound of the objective function of (10), we have an optimality cut of the form

$$\theta - \pi_1^T(b'' - Sx'^*) + \pi_2^T u'' - \pi_3^T l'' \geq 0$$

Creating an Advanced Basis

Let $x^k = \left((x')^k, (x'')^k\right)$ denote the solution vector of the master and subproblem in the kth pass

- This solution may be infeasible, feasible, or feasible and optimal for the original LPEN problem
- Create the starting basis as follows
  - The variable x_i^k for the ith component appears as a basic variable in the solution of the master or subproblem $\Rightarrow$ basic
  - If not $\Rightarrow$ non-basic, analyze the solution values
    - $x_i^k = l_i \Rightarrow$ non-basic at lower bound
    - $x_i^k = u_i \Rightarrow$ non-basic at upper bound
- The basis factorization procedure INVERT uses this info to create an initial factorization of this basis as a simplex starting point for the LPEN problem

Conclusions and Future Directions

- It is shown that exploiting the embedded network structure significantly improves the performance of the simplex algorithm
  - See the paper for the computational results and comparisons among different procedures
- Refinement of decomposition methods will most likely lead to further improvement of the procedure...