Elliot F. Poirier
Summer 2024

McGill University
MATH470 - Honours Research Project

Notes on Dantzig-Wolfe Decomposition

1. Block-Angular Structure

A matrix A has a **block angular structure** if it is of this form:

$$A = \begin{bmatrix} A_1 & A_2 & \cdots & A_{n-1} & A_n \\ A_{n+1} & 0 & \cdots & 0 & 0 \\ 0 & A_{n+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & A_{2n} \end{bmatrix}$$

where A_i are submatrices of size $m_i \times n_i$. We call the first n matrices $A_1, A_2, \cdots, A_{n-1}$ the bounding constraints and the last n matrices the independent constraints. Notice that this easily becomes a linear program that looks like this,

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \vec{c}_i \vec{x}_i \\ \text{subject to} \quad & [A_1, \cdots, A_n, I]\vec{x}^T = \vec{b}_0 \quad \text{linking constraints} \\ & A_{n+i}\vec{x}_i = \vec{b}_i, \quad \forall i \in \{1, \cdots, n\} \quad \text{independent constraints} \\ & \vec{x}^T \geq 0, \quad \vec{x}_i \geq 0, \quad \forall i \in \{1, \cdots, n\} \quad \text{non-negativity constraints} \end{aligned}$$

In this linear program, the $\vec{c}_i$ are the costs and $\vec{x} = [\vec{x}_1, \cdots, \vec{x}_n]^T$ are the decision variables and $\vec{b}_i$ are the right hand side of the constraints. The aim of the Dantzig-Wolfe decomposition is to be more computationally efficient in solving this problem. We start with a **restricted master problem** (RMP), which usually encompasses all the linking constraints and then check the independent constraints one by one by solving them as subproblems.

2. Dantzig-Wolfe Decomposition

2.1 Intro

We'll start by giving a high level view of the algorithm, which is basically doing these steps (Wikipedia):

1. Start with a feasible solution to the RMP, formulate new objective functions for each subproblem such that the subproblems will offer solutions that improve the current objective of the master program.

2. Re-solve the subproblems given their new objective functions, the optimal value for each subproblem is offered to the master program.

3. Incorporate one or all of the new columns generated by the solutions to the subproblems into the master program based on their respective ability to improve the original problem's objective.

4. Perform x iterations of the simplex algorithm in the master program, where x is the number of columns incorporated.

5. If the objective is improved, go to step 1. Otherwise, we ran out of improvement so we return the solution.

Note that Dantzig-Wolfe is most effective when the subproblems are easy to solve. Now, let's break it down into more detail. The first step consists of solving the RMP and then defining the objective functions. The RMP looks like this,

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \vec{c}_i \vec{x}_i \\ \text{subject to} \quad & [A_1, \cdots, A_n, I]\vec{x}^T = \vec{b}_0 \\ & \vec{x}^T \geq 0 \end{aligned}$$

The subproblems look like this, for every ith subproblem,

$$\begin{aligned} \text{maximize} \quad & \vec{c}_i \vec{x}_i \\ \text{subject to} \quad & A_{n+i}\vec{x}_i = \vec{b}_i \\ & \vec{x}_i \geq 0 \end{aligned}$$

What interests us is the feasible region of the subproblems, we will denote them as $S_i = \{\vec{x}_i \geq 0 | A_{n+i}\vec{x}_i = \vec{b}_i\}$. Assume that S_i is bounded for simplicity (it can easily be extended to having unbounded regions, we would consider unbounded rays). We have a set of vertices $V_i = \{(v_{i,1}, \cdots, v_{i,n}) | v_{i,j} \in S_i\}$ which completely defines the feasible region of the subproblem. This implies that we can represent every $x_i \in S_i$ as a convex combination of the vertices $v_{i,j}$, i.e. $\exists\{\lambda_{i,j} \geq 0\}_j$ s.t. $x_i = \sum_{j=1}^{n} \lambda_{i,j} v_{i,j}$ and with $\sum_{j=1}^{n} \lambda_{i,j} = 1$ (this follows from the representation theorem).

Now, we can transform the RMP by using the $\lambda_{i,j}$ as the new decision variables.

$$\begin{array}{lllr} \min & \sum_{i=1}^{m} \sum_{j=1}^{N_i} \lambda_{i,j} \left(c_i v_{i,j}\right) & & \\ \text{s.t.} & \sum_{i=1}^{m} \sum_{j=1}^{N_i} \lambda_{i,j} \left(A_{0,i} v_{i,j}\right) & = b_0 & \\ & \sum_{j=1}^{N_i} \lambda_{i,j} & = 1 & \forall i = 1, 2, \ldots, m \\ & \lambda_{i,j} & \geq 0 & \forall i, j \end{array}$$

Notably, we reduced the number of constraints to $m + m_0$, since there aren now m independent constraints and m_0 linking constraints. Basically, we squashed all the independent constraint blocks into single constraints. However, there are now way more variables, which is why we will use an altered simplex method.

2.2 Altered Simplex

We start by creating the dual variables, $y_0 \in \mathbb{R}^{m_0}$ and $z \in \mathbb{R}^m$ which set the sum $\sum_{i=1}^{m} \lambda_{i,j} = 1$. Note that each column represents one vertex. And so, we want to find out which columns have negative reduced cost. If we consider the column for $\lambda_{i,j}$, then the reduced cost is

$$\bar{c}_i = c_i v_{i,j} - y_0 A_{0,i} v_{i,j} - z_i$$

which we can obtain by taking the inner product of the dual variables with the column for $\lambda_{i,j}$.

$$\begin{bmatrix} y_0 & z \end{bmatrix} \begin{bmatrix} A_{0i} v_{ij} \\ \hline c_i \end{bmatrix} \quad \text{(unsure about this)}$$

So now we want to know if there exists $v_{i,j}$ such that $\bar{c}_i < 0$ or equivalently, $c_i v_{i,j} - y_0 A_{0,i} v_{i,j} - z_i < 0$. If so, we can then find the column for $\lambda_{i,j}$ that has the smallest reduced cost? This means we want to find a vertex of S_i

such that $(c_i - y_0 A_{0,i})v_{i,j} - z_i < 0$.

We now consider the resulting subproblem,

$$\begin{array}{lll} \min & \bar{c}_i x_i & \\ \text{s.t.} & A_{n+i} x_i & = b_i, \\ & x_i & \geq 0 \end{array}$$

Since S_i is nonempty (otherwise the master problem is infeasible), this problem must contain a vertex $\mathbf{v}$ which is the optimal solution. If $\bar{c}_i\mathbf{v} < z_i$ then we know that $\mathbf{v}$ is a negative cost vertex and we should add it to the basis. If this is false $\forall i$, then no column of S_i has negative reduced cost. Hence, after solving the subproblems for all S_i, we either find a negative reduced cost column or otherwise we show that the current solution to the master problem is optimal.

Notice that this in itself is fundamentally a revised version of the simplex algorithm, however with significantly reduced usage of space. Additionally, it is more efficient computationally since the slave problems can be solved in parallel (that is the big advantage), we simply choose the first negative reduced cost column subproblem to be computed.

2.3 Examples

Our particular interest with this algorithm stems from the particular structure of the optimization problem we are trying to solve. Let us formalize the problem of block allocation (mining blocks) in the context of mine planning. Our particular problem is of this form,

$$\begin{aligned} &\max \sum_{b\in B} \left(V_{b,A,I} m_{b,A,I} + V_{b,A,II} m_{b,A,II} + V_{b,B,I} m_{b,B,I} + V_{b,B,II} m_{b,B,II}\right) \\ &\text{st.} \quad \sum_{b\in BB} \left(\frac{m_{b,A,I}}{r_{A,I}} + \frac{m_{b,A,II}}{r_{A,II}} + \frac{m_{b,B,I}}{r_{B,I}} + \frac{m_{b,B,II}}{r_{B,II}}\right) \leqslant t_{\text{period}} \quad (1) \\ &\quad m_{b,A,I} + m_{b,B,I} \leqslant m_{b,I} \\ &\quad m_{b,A,I} + m_{b,B,I} \leqslant m_{b,I} \quad \forall b \in \mathcal{B} \quad (3) \\ &\quad m_{ijk}, r_{jk} \geqslant 0 \quad \forall i \in \mathcal{B}, j \in \text{ modes}, k \in \text{ ores} \end{aligned}$$

With (1) being that total processing time cannot be greater than the total period (2) being that the total amount of ore being processed in a block cannot exceed the capacity of the block. It is to be noted that we are missing a constraint related to the rate of extraction.

Now, we will look at two examples to illustrate this, both with two modes (A, B) and two ore types (I, II), one with one block, and one with two blocks.

2.3.1 With one block, we have this problem,

$$
\begin{aligned}
&\max\left(V_{A,I}m_{A,I} + V_{A,II}m_{A,II} + V_{B,I}m_{B,I} + V_{B,II}m_{B,II}\right)\\
&\text{st. } \frac{m_{A,I}}{r_{A,I}} + \frac{m_{A,II}}{r_{A,II}} + \frac{m_{B,I}}{r_{B,I}} + \frac{m_{B,II}}{r_{B,II}} \leqslant t_{\text{period}} \quad (1)\\
&\qquad m_{A,I} + m_{B,I} \leqslant m_I\\
&\qquad m_{A,I} + m_{B,I} \leqslant m_I\\
&\qquad m_{A,I}, m_{A,II}, m_{B,I}, m_{B,II}, r_{A,I}, r_{A,II}, r_{B,I}, r_{B,II} \geqslant 0
\end{aligned}
$$

So, now our first step,

2.3.2 With two blocks, we have this problem,

$$
\begin{aligned}
&\max\left(V_{1,A,I}m_{1,A,I} + V_{1,A,II}m_{1,A,II} + V_{1,B,I}m_{1,B,I} + V_{1,B,II}m_{1,B,II}\right) + \left(V_{2,A,I}m_{2,A,I} + V_{2,A,II}m_{2,A,II}\right.\\
&\text{st. } \left(\frac{m_{1,A,I}}{r_{1,A,I}} + \frac{m_{1,A,II}}{r_{1,A,II}} + \frac{m_{1,B,I}}{r_{1,B,I}} + \frac{m_{1,B,II}}{r_{1,B,II}}\right) + \left(\frac{m_{2,A,I}}{r_{2,A,I}} + \frac{m_{2,A,II}}{r_{2,A,II}} + \frac{m_{2,B,I}}{r_{2,B,I}} + \frac{m_{2,B,II}}{r_{2,B,II}}\right) \leqslant t_{\text{period}}\\
&\qquad m_{1,A,I} + m_{1,B,I} \leqslant m_I\\
&\qquad m_{1,A,II} + m_{1,B,II} \leqslant m_{II}\\
&\qquad m_{2,A,I} + m_{2,B,I} \leqslant m_I\\
&\qquad m_{2,A,II} + m_{2,B,II} \leqslant m_{II}\\
&\qquad m_{i,j,k}, r_{j,k} \geqslant 0 \quad \forall i \in \{1,2\} \quad j \in \{A,B\} \quad k \in \{I,II\}
\end{aligned}
$$