An Introduction to Branch-and-Price
Part I: Theory

Adam N. Letchford

Department of Management Science
Lancaster University

September 2011

Outline

1. The basic idea
2. Dantzig-Wolfe decomposition
3. The resulting lower bound
4. Exploiting block-angular structure
5. The case of identical blocks

The basic idea

Let's recall how branch-and-cut works:

- We formulate a COP as an ILP.
- We derive a stronger formulation with a huge number of constraints.
- We can't include all of the constraints in the LP...
- ... so we start with a few and generate others as needed.
- To generate constraints we need separation algorithms.
- After adding new constraints, we re-optimise with dual simplex.
- When we can't find any more constraints, we branch.

The basic idea (cont.)

Branch-and-price is a 'dual' version of branch-and-cut:

- We formulate a COP as an ILP.
- We derive a stronger formulation with a huge number of *variables*.
- We can't include all of the *variables* in the LP...
- ... so we start with a few and generate others as needed.
- To generate variables we need *pricing* algorithms.
- After adding new variables, we re-optimise with *primal* simplex.
- When we can't find any more variables, we branch.

The basic idea (cont.)

Question: How can we strengthen an ILP formulation by increasing the number of variables?!

Answer: We use the decomposition technique of Dantzig & Wolfe (1960).

(For simplicity, I'll concentrate on 0-1 LPs.)

Dantzig-Wolfe decomposition

Suppose we have a 0-1 LP of the form:

$$\min \{c \cdot x : Ax \geq b,\ Cx \geq d,\ x \in \{0,1\}^n\}.$$

Suppose also that the constraints $Ax \geq b$ are 'nice', whereas the constraints $Cx \geq d$ are 'nasty'.

That is, we could solve the 0-1 LP quickly (somehow) if the constraints $Cx \geq d$ were not present.

Dantzig-Wolfe decomposition (cont.)

One way to handle 0-1 LPs of this type is to remove the 'nasty' constraints from the problem, but modify the objective function.

This is Lagrangian relaxation, which you have already seen.

An alternative (which usually works just as well or even better) is to use Dantzig-Wolfe decomposition.

Dantzig-Wolfe decomposition (cont.)

We define the two polyhedra:

$$P^1 = \{x \in [0,1]^n : Ax \geq b\}$$

$$P^2 = \{x \in [0,1]^n : Cx \geq d\} .$$

The 0-1 LP can now be written as:

$$\min \left\{ c \cdot x : x \in P^1 \cap P^2, \; x \in \{0,1\}^n \right\} .$$

Dantzig-Wolfe decomposition (cont.)

Now consider the integral hull of P^1:

$$P_I^1 = \text{conv}\left\{x \in \{0,1\}^n : Ax \geq b\right\}.$$

By definition, P_I^1 is the convex hull of a finite number of vertices.

Let $v^1, \ldots, v^t \in \{0,1\}^n$ be these vertices.

Note: the number of vertices can be huge (exponential).

Dantzig-Wolfe decomposition (cont.)

Any point in P_I^1 can be written as a *convex combination* (weighted average) of the vertices $v^1, \ldots, v^t$.

More formally, if $x^* \in P_I^1$, then there exist non-negative multipliers $\lambda_1, \ldots, \lambda_t$, summing to one, such that:

$$x^* = \sum_{k=1}^{t} \lambda_k v^k.$$

Dantzig-Wolfe decomposition (cont.)

This leads to the following 0-1 LP formulation of our problem:

$$\begin{aligned} \min \quad & c \cdot x \\ \text{s.t.} \quad & Cx \geq d \\ & x = \textstyle\sum_{k=1}^{t} v^k \lambda_k \\ & \textstyle\sum_{k=1}^{t} \lambda_k = 1 \qquad (*) \\ & \lambda \in \{0,1\}^t \\ & x \in \{0,1\}^n. \end{aligned}$$

This 0-1 LP, called the *master* problem, can have a huge number of variables (one for each vertex of P_I^1).

The constraint (*) is called a *convexity constraint*.

Dantzig-Wolfe decomposition (cont.)

It is usual practice to eliminate the x variables, so that the master problem takes the form:

$$\begin{array}{ll} \min & \tilde{c} \cdot \lambda \\ \text{s.t.} & \tilde{C}\lambda \geq d \\ & \sum_{k=1}^{t} \lambda_k = 1 \quad (*) \\ & \lambda \in \{0,1\}^t. \end{array}$$

Here, $\tilde{c}_k$ denotes the cost of vertex v^k and $\tilde{C}_{kj}$ denotes the 'contribution' of vertex v^k to the jth constraint in the system $Cx \geq d$.

The resulting lower bound

- So that's how we convert a 0-1 LP into a different 0-1 LP with a huge number of variables.
- I'll explain how to actually *solve* the master problem in the next session.
- For now, let's try to understand why the master problem is a 'stronger' formulation than the original problem.
- To do that, we look at the lower bounds that we get when we solve the LP relaxations of the two problems.

The resulting lower bound (cont.)

The LP relaxation of the original problem is:

$$\min \left\{ c \cdot x : x \in P^1 \cap P^2 \right\}.$$

Geoffrion (1974) showed that solving the LP relaxation of the master problem is equivalent to solving:

$$\min \left\{ c \cdot x : x \in P_I^1 \cap P^2 \right\}.$$

So, if $P_I^1 \neq P^1$, the LP relaxation of the master problem will give a better bound than that of the original problem.

The resulting lower bound (cont.)

Geoffrion (1974) also showed that solving the LP relaxation of the master problem gives the *best possible* lower bound that could be obtained with Lagrangian relaxation.

That is, if we used *the best possible* multipliers in a Lagrangian relaxation, we would get the same bound.

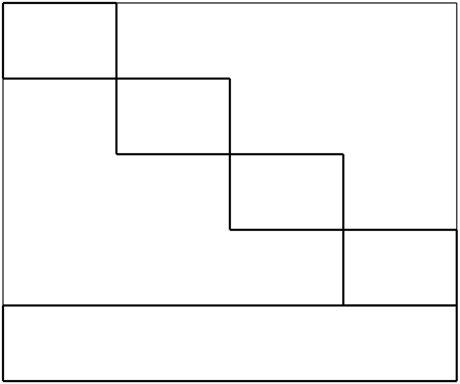
This is what makes Dantzig-Wolfe decomposition attractive.

Block-angular structure

Dantzig-Wolfe decomposition is particularly effective when applied to ILPs that have a *block-angular* constraint matrix.

This means that, if the 'nasty' constraints $Cx \geq d$ are deleted, the resulting constraint system $Ax \geq b$ decomposes into a number of much smaller systems ('blocks').

Block-angular structure (cont.)



‘Nice’ system

$Ax \geq b$

‘Nasty’ system

$Cx \geq d$

Block-angular structure (cont.)

- In this situation, we can substantially reduce the number of variables in the master problem.
- Instead of one integral hull P_I^1, we have one integral hull for each block.
- Since each integral hull involves a much smaller number of variables, it is likely to have a much smaller number of vertices.
- In the master problem, we have one set of λ variables per block and one convexity constraint per block.

Block-angular structure (cont.)

This is the resulting master:

$$
\begin{array}{lll}
\min & \sum_{j=1}^{s} \tilde{c}^j \cdot \lambda^j & \\
\text{s.t.} & \sum_{j=1}^{s} \tilde{C}^j \lambda^j \geq d & \\
& \sum_{k=1}^{t_j} \lambda_k^j = 1 & (j = 1, \ldots, s) \\
& \lambda^j \in \{0,1\}^{t_j} & (j = 1, \ldots, s).
\end{array}
$$

Here, s is the number of blocks, t_j is the number of vertices of the jth integral hull and v^{jk} is the kth vertex of the jth integral hull.

Block-angular structure (cont.)

Lots of block-angular ILPs arise in Combinatorial Optimisation!

For example:

- one block for each facility / depot / factory
- one block for each worker / machine / vehicle
- one block for each product type / commodity

The case of identical blocks

- Moreover, in Combinatorial Optimisation it often happens that the blocks are *identical*.
- E.g., the machines/vehicles may all be of the same type.
- In this case, all of the integral hulls are the same.
- So, we need only one set of λ variables.
- This *dramatically* reduces the number of variables (though it can still grow exponentially).

The case of identical blocks (cont.)

This is the resulting master:

$$\begin{array}{ll} \min & \tilde{c} \cdot \lambda \\ \text{s.t.} & \tilde{C}\lambda \geq d \\ & \sum_{j=1}^{t} \lambda_j = s \quad (*) \\ & \lambda \in \{0,1\}^t. \end{array}$$

It looks just like the master for the general case (not block-angular), except that the convexity constraint now has a right-hand side of s (the number of blocks).

SUMMARY

- Dantzig-Wolfe decomposition is a well-known technique for reformulating ILPs.
- It is best used when the ILP has block-angular structure...
- ... especially if all of the blocks are identical.
- Such ILPs arise frequently in Combinatorial Optimisation.
- In the next session, I will explain how to solve the master problem using the *branch-and-price* method.
- I'll also give some examples of COPs that can be solved quickly using branch-and-brice.