4
Decomposition Algorithms

4.1 Summary

Dating back from the start of the Mathematical Programming area, Decomposition Algorithms have been introduced as a resort to handle large scale Linear Programs (LP). However, LP codes have evolved enormously, and together with advances in computer hardware, they are able nowadays to cope with almost all LP problems found in practice. Thus, the interest in using decomposition methods in the last two decades has switched to Mixed Integer Linear Programming (MILP) problems.

Ever since the seminal papers of Dantzig and Wolfe [Dan60] and Benders [Ben62], the literature has flourished with theoretical extensions and practical applications of these two basic approaches. In the real scheme of things we can say that almost every method used to solve MIP problems is based on a decomposition strategy. And this comes with no surprise since one of the most basic ideas for solving large problems, that is so well exploited in Computer Science, is "Divide and Conquer". Even traditional methods to solve MIP problems, such as Branch-and-Bound and Cutting Planes, can be faced as decomposition algorithms, as they, in fact, split a problem in its linear relaxation and integrality constraints.

In this chapter we focus on the traditional decomposition methods, i.e., Dantzig and Wolfe, Benders decomposition and Lagrangean relaxation. This will allow us to make a connection with the novel decomposition algorithm presented in Chapter 5.

PUC-Rio - Certificação Digital Nº 0611954/CA

4.2 Dantzig and Wolfe Decomposition

Dantzig and Wolfe decomposition is an algorithm that was designed originally to solve large LP with special structures [Dan60]. The basic idea of this technique is to apply the Minkowsky Theorem or the Representation Theorem for convex polyhedra (see Theorem 4.1) to a subset of constraints of a given problem, where its extreme points and extreme rays are generated on the fly.

Suppose for example we have the LP,

$$\begin{aligned} \min \quad & f^T x \\ s.t. \quad & Ax \geq b \\ & Cx \geq d \\ & x \geq 0 \end{aligned} \tag{4.1}$$

where f^T, x, b and d are vectors, A and B are matrices, all of conformable dimensions. Let us take the inequalities $Cx \geq d$ as the subset of constraints to be considered implicitly.

As mentioned above, one of the key element of this decomposition algorithm is the Minkowsky Theorem that is stated without proof in the following.

Theorem 4.1 (Minkowsky) *If P is the polyhedron $\{x \mid Cx \geq d\}$, any x that satisfies $Cx \geq d$ is the sum of*

a) A convex combination of extreme points of P

b) And a conical combination of extreme rays of P

Using this Theorem, every x that satisfies $Cx \geq d$ can be written as

$$\begin{aligned} & x = \sum_{k \in K} \lambda_k y^k + \sum_{l \in L} \mu_l w^l \\ & \sum_{k \in K} \lambda_k = 1 \\ & \lambda_k, \mu_l \geq 0 \end{aligned} \tag{4.2}$$

Substituting $Cx \geq d$ for the expressions in (4.2), the problem (4.1) becomes

$$\begin{aligned} \min \quad & f^T x \\ s.t. \quad & Ax \geq b \\ & x = \sum_{k \in K} \lambda_k y^k + \sum_{l \in L} \mu_l w^l \\ & \sum_{k \in K} \lambda_k = 1 \\ & x \geq 0, \lambda_k \geq 0, \mu_l \geq 0, k \in K, l \in L \end{aligned} \tag{4.3}$$

PUC-Rio - Certificação Digital Nº 0611954/CA

Now substituting out the x variables, we write

$$
\begin{array}{lll}
 & \min & \sum_{k \in K} \lambda_k f^T y^k + \sum_{l \in L} \mu_l f^T w^l \\
(DW) & s.t. & \sum_{k \in K} \lambda_k A y^k + \sum_{l \in L} \mu_l A w^l \geq b \qquad (4.4) \\
 & & \sum_{k \in K} \lambda_k = 1 \qquad (4.5) \\
 & & \lambda_k \geq 0, \mu_l \geq 0, k \in K, l \in L
\end{array}
$$

Formulation (DW) is called the Dantzig and Wolfe Master program, where constraints (4.4) are the coupling constraints and constraint (4.5) is the convexity constraint. This formulation alone does not make the problem easier because (DW) can have an enormous number of extreme points and extreme rays. Let us suppose the dimension of matrix A equals to $m_1 \times n$ and the dimension of matrix C equals to $m_2 \times n$, then the dimension of the original problem is $(m_1+m_2)\times n$, while the dimension of (DW) is $(m_1+1)\times(|K|+|L|)$. Thus, although (DW) has a smaller number of constraints, its number of variables can be huge since the number of extreme points and extreme rays of a polyhedron can be very large. To use this idea to effectively solve large scale LP problems, we need to avoid considering all extreme points and extreme rays of $Cx \geq d$. This is when the idea of column generation comes into play, i.e., we start by including only a few number of extreme points and extreme rays in the problem and we add more on the fly in a as needed basis.

(a) Column Generation

To cope with solving (DW) we borrow the ideas from the revised simplex algorithm. We know that to find a basic solution to (DW) we need at least $m_1 + 1$ linear independent columns. However, to check whether or not this solution is also optimal we must prove that there is no other nonbasic variable/column that is worth coming into the basis. Therefore, we start the algorithm constructing a Restricted Dantzig and Wolfe Master program (RDW) with a subset of columns from (DW) and additional ones are generated as the algorithm proceeds.

PUC-Rio - Certificação Digital Nº 0611954/CA

$$
(RDW) \quad \begin{array}{ll} \min & \sum_{k\in\Lambda} \lambda_k f^T y^k + \sum_{l\in\Omega} \mu_l f^T w^l \\ s.t. & \sum_{k\in\Lambda} \lambda_k A y^k + \sum_{l\in\Omega} \mu_l A w^l \geq b \qquad (4.6) \\ & \sum_{k\in\Lambda} \lambda_k = 1 \qquad (4.7) \\ & \lambda_k \geq 0, \mu_l \geq 0, k \in \Lambda, l \in \Omega \end{array}
$$

where Λ is a small subset of K and Ω a small subset of L.

Let us assume we have solved problem (RDW). In order to verify whether the current solution can be improved, we need to check if some variable that is not yet in the problem has negative reduced cost. To do this we proceed in two steps. The first step is to try to find some variable associated with an extreme point. Recalling from Linear Programming Theory that, in general, the reduced cost of a variable x_j is $c_j - u^T A_j$, where u is the vector of dual variables, and A_j is the column corresponding to x_j. Let (u, α) be the vector of dual variables relative to (4.6) and (4.7), respectively. Then the reduced cost of a variable that is not yet in the problem is

$$
f^T y^k - u^T A y^k - \alpha \tag{4.8}
$$

Now we can check if some variable λ_k has a negative reduced cost by minimizing (4.8) over all points in $Cx \geq d$

$$
\begin{array}{ll} \min & f^T y - u^T A y - \alpha \\ s.t. & Cy \geq d \\ & y \geq 0 \end{array} \tag{4.9}
$$

If the solution of (4.9) is negative, then we have found a candidate associated with an extreme point to be included in (RDW). Let y^* be the extreme point solution of (4.9), we add to (RDW) a new variable λ_k with column $(Ay^*, 1)$ and cost $f^T y^*$.

The next step is to check if some variable μ_l has negative reduced cost. In the same way as proceeded above, we need to solve the following subproblem

$$
\begin{array}{ll} \min & f^T w - u^T A w \\ s.t. & Cw \geq d \\ & w \geq 0 \end{array} \tag{4.10}
$$

PUC-Rio - Certificação Digital Nº 0611954/CA

Note, however, that the only difference from (4.9) is the constant term $-\alpha$, and then we do not need to solve this problem again. In fact, by Linear Programming, we know that for an extreme ray the term $f^T w^k - u^T A w^k$ goes to $-\infty$. Thus, the way to verify whether or not there is a extreme ray to be added to the master problem is to check if problem (4.9) is unbounded. If this is the case, let w^* be the extreme ray solution of (4.10), we add to (RDW) a new variable μ_l with column $(Aw^*, 0)$ and cost $f^T w^*$.

At this point we solve (RDW) with the added columns and repeat the process, until there are no more improving columns, which is to say that the optimal value of subproblems (4.9) and (4.10) are non-negative. An outline of this method is presented by Algorithm 1.

In this work we do not go into the details of the computational issues regarding the implementation of this algorithm. There are many practical points that need to be addressed if one wants to solve a problem using this method. Besides, some aspects of its implementation is in most cases problem dependent, requiring the mathematical programming specialists to use all their knowledge to effectively solve it.

Finally, we have presented an algorithm to solve LP problems. Nevertheless, in the last two decades this algorithm has been used as a powerful tool to obtain tighter relaxation for Mixed Integer Programming problems. Another line of research reported in the literature is to combine the key ingredients of this decomposition algorithm with the Branch-and-Cut method to derive a new procedure called Branch-and-Cut-and-Price [Bar96]. We can find many papers in the literature where this idea has proved to be effective in solving some large scale MIP problems [Bar00, Bor06, Fuk06, Uch08, Rop09].

PUC-Rio - Certificação Digital Nº 0611954/CA

Algorithm 1 Dantzig Wolfe Decomposition

Step 0: Find initial sets of columns Λ and Ω and construct the Restricted Dantzig and Wolfe Master problem (RDW). Assume the restricted problem is feasible.
Step 1: Optimize the restricted master problem (RDW) to obtain an optimal set of dual variables (u, α).
Step 2: Solve the column generation subproblem (4.9)
if $f^T y^k - u^T A y^k - \alpha$ goes to $-\infty$ **then**
  Add the column $(Aw^*, 0)$ and cost $f^T w^*$ to the restricted master problem (RDW).
  Go to step 1.
else if $f^T y^k - u^T A y^k - \alpha < 0$ **then**
  Add the column $(Ay^*, 1)$ and cost $f^T y^*$ to the restricted master problem (RDW).
  Go to step 1.
else
  Stop.
end if

PUC-Rio - Certificação Digital Nº 0611954/CA

4.3 Benders Decomposition

Differently from the Dantzig and Wolfe decomposition, Benders decomposition [Ben62] was introduced in the early sixties as an algorithm to solve MILP programs. However, it does not mean that we can not apply it to solve LP programs. In fact, provided that the LP problem is well-structured, it might even be employed with advantage to solve LP programs over traditional methods. One important distinction from the Dantzig and Wolfe decomposition is the way in which the problem is decomposed. While Dantzig and Wolfe splits the set of constraints into two subsets, the Benders decomposition divides the variable set into two subsets. This characteristic suggests some connections between these two methods, and as matter of fact, in the literature Benders decomposition is often described as Dantzig and Wolfe decomposition applied to the dual of a problem. In reality, it can be shown that in the context of LP programs they are dual of each other. In the sequel we explain how this method is used in the case of MILP problems.
Let us consider the following MILP

$$\begin{array}{ll} \min & f^T x + g^T y \\ s.t. & Ax + By \geq c \\ & x \geq 0, y \in \mathbb{Z} \end{array} \tag{4.11}$$

where x and y are vectors of continuous and integer variables, respectively, with dimensions p and q. Again, f, g, c are vectors and A, B are matrices of appropriate dimensions.

Suppose we have an oracle to determine a trial value for y. Let us fix the value y to this initial guess $\bar{y}$, so we obtain the following subproblem

$$\begin{aligned} \min \quad & f^T x + g^T \bar{y} \\ s.t. \quad & Ax \geq c - B\bar{y} \\ & x \geq 0 \end{aligned} \tag{4.12}$$

The dual of subproblem (4.12), ignoring the constant term $g^T\bar{y}$, is written as

$$\begin{aligned} \max \quad & u^T(c - B\bar{y}) \\ s.t. \quad & u^T A \leq f \\ & u \geq 0 \end{aligned} \tag{4.13}$$

Now, let us suppose that the optimal solution value of (4.12) is z^* and the optimal solution of its dual (4.13) is u^*. Thus, by strong LP duality

$$z^* - g^T\bar{y} = (u^*)^T(c - B\bar{y}) \tag{4.14}$$

Furthermore, observe that the feasible region of the dual problem (4.13) does not depend on the value of $\bar{y}$. Therefore, u^* is always a feasible solution to (4.13) disregarding the value of $\bar{y}$ in its objective function. Hence, by weak LP duality

$$z \geq g^T y + (u^*)^T(c - By) \tag{4.15}$$

The inequality (4.15) is a lower bound on the objective function of the original problem (4.11). This is the key element used in Benders decomposition to generate valid Benders cuts to the master problem (to be defined later). It should be noted that the cut (4.15) says that if we set $y = \bar{y}$ again, the resulting objective function value will be at least the value $\bar{z}$ we just obtained in the subproblem, since $z \geq g^T\bar{y} + (u)^T(c - B\bar{y})$. It also gives us a lower bound for other values of y we might try.

Now let us consider the case where the dual subproblem (4.13) is unbounded. In this situation the following inequality holds

$$(u^*)^T(c - By) > 0 \tag{4.16}$$

where now $(u^*)^T$ is an extreme ray solution.

Our goal is to eliminate this extreme ray from future solutions. To do so, we have to ensure that the term $(u^*)^T(c - By)$ is less than or equal to zero in all future steps. Thus, the Benders feasibility cut to be added to the master problem must be

$$(u^*)^T(c - By) \leq 0 \tag{4.17}$$

To calculate the next trial value of y, we solve the following master problem (4.18) obtained by appending all Benders and infeasibility cuts generated so

PUC-Rio - Certificação Digital Nº 0611954/CA

far

$$
\begin{aligned}
\min \quad & z \\
s.t. \quad & z \geq g^T y + (u^k)^T (c - By), \quad k \in K \\
& (u^k)^T (c - By) \leq 0, \quad k \in L \\
& y \in \mathbb{Z}
\end{aligned}
\tag{4.18}
$$

where K is the set of extreme points and L the set of extreme rays for the subproblem.

We should notice that the subproblem (4.12) is a restriction on the original problem (4.11), hence its optimum solution value z^* is a upper bound on the original problem objective function. Also, the master problem (4.18) is a relaxation of the original problem (4.12) since we do not consider all constraints from every extreme points and extreme rays of the subproblem. Thus, the optimum solution value of the master problem provides a lower bound on the original problem objective function. This is an interesting feature of this algorithm because one can terminate it prematurely if the lower and upper bounds found are close enough. An outline of this method is presented by Algorithm 2.

PUC-Rio - Certificação Digital Nº 0611954/CA

Algorithm 2 Benders Decomposition

Step 0: Find an initial guess $\bar{y} \in \mathbb{Z}$ and set the upper bound $UB = \infty$ and the lower bound $LB = -\infty$.

Step 1: Solve the Benders' subproblem,

$$\begin{aligned} \max \quad & u^T(c - B\bar{y}) \\ s.t. \quad & u^T A \leq f^T \\ & u \geq 0 \end{aligned}$$

if The subproblem is infeasible **then**

  Terminate.

else if The subproblem has an optimum solution **then**

  Set UB $\leftarrow \min(UB, g^T y + (u^k)^T(c - By))$

  Generate a Benders cut $z \geq g^T y + (u^k)^T(c - By)$ and add it to the master problem.

else if The subproblem is unbounded **then**

  Generate a Benders feasibility cut $(u^k)^T(c - By) \leq 0$ and add it to the master problem.

end if

Step 3: Solve the Benders' master problem,

$$\begin{aligned} \min \quad & z \\ s.t. \quad & z \geq g^T y + (u^k)^T(c - By), \quad k \in K \\ & (u^k)^T(c - By) \leq 0, \quad k \in L \\ & y \in \mathbb{Z} \end{aligned}$$

to find a new value $\bar{y}$ and a solution value z^*.

Set LB $\leftarrow \max(LB, z^*)$

if $UB - LB < \epsilon$ **then**

  Stop, otherwise go to Step 1.

end if

PUC-Rio - Certificação Digital Nº 0611954/CA

4.4 Lagrangean Relaxation

The idea behind Lagrangean relaxation was first introduced by Held and Karp [Hel70, Hel71] in the context of finding better relaxations for the Traveling Salesman Problem, however the term Lagrangean relaxation was only coined by Geoffrion [Geo74] in a paper four years later. This method was specially designed to provide strong relaxations for some large scale integer programming problems [Geo74, Fis81], but it is not limited to this kind of applications. In this section we explain the basic ideas of the Lagrangean relaxation and show its most important properties.

Consider the linear integer programming

$$
\begin{array}{llll}
 & \max & f^T x & \\
(IP) & s.t. & Ax \leq b & (4.19)\\
 & & Cx \leq d & (4.20)\\
 & & x \in \{0,1\}^n &
\end{array}
$$

where, f^T, x, b and d are vectors, A and B are matrices, all of conformable dimensions.

Let us suppose that the set of constraint $Ax \leq b$ is complicating, in the sense that if one removes it from (IP), the problem becomes easier to solve. One can take advantage of this information to construct the following problem

$$
\begin{array}{llll}
 & \max & f^T x + \lambda(b - Ax) & \\
(LR_\lambda) & s.t. & Cx \leq d & (4.21)\\
 & & x \in \{0,1\}^n &
\end{array}
$$

The problem (LR_λ) is called the Lagrangean relaxation of (IP) and $\lambda \geq 0$ is the vector of lagrangean multipliers relative to the constraints $Ax \leq b$. It is clear that (LR_λ) is a relaxation of (IP) since its objective function $f^T x + \lambda(b - Ax)$ is always greater than $f^T x$ for every feasible solution of (IP), moreover the feasible region of (IP) is contained in the feasible region of (LR_λ).

We should note that the optimum solution of (LR_λ), $v(LR_\lambda)$, is dependent on the value of λ. Then, we can try to find a λ that provides the best upper bound possible. The problem of finding the best relaxation (LR_λ) of (IP) is called the lagrangean dual and it is defined as

PUC-Rio - Certificação Digital Nº 0611954/CA

$$(LD) \quad min_{\lambda \geq 0} v(LR_\lambda) \tag{4.22}$$

Observe that $v(LR_\lambda)$ is a convex piece-wise function and the problem (4.22) is usually solved using a subgradient algorithm (see, for instance, [Gui03]). A complete overview on algorithms to solve this lagrangean dual problem can be found in Guta [Gut03].

In the following, we will state and prove a theorem that brings up important properties of the Langragean relaxation.

Theorem 4.2 *The Lagrangean dual* (LD) *is equivalent to the primal relaxation*

$$\begin{array}{lll} & \max & f^T x \\ (PR) & s.t. & Ax \leq b \\ & & x \in Co\{x \in \{0,1\}^n \,|Cx \leq d\} \end{array}$$

in the sense that $v(LD) = v(PR)$*, where* $Co\{S\}$ *stands for the convex hull of points of* S*.*

Proof:

Let us call $Co\{x \in \{0,1\}^n \,|Cx \leq d\} = \tilde{C}x \leq \tilde{d}$, then (PR) can be written as

$$\begin{array}{lll} & \max & f^T x \\ (\widetilde{PR}) & s.t. & Ax \leq b \\ & & \tilde{C}x \leq \tilde{d} \\ & & x \geq 0 \end{array}$$

By linear duality,

$$\begin{array}{lll} & \min & \lambda^T b + \beta^T \tilde{d} \\ (\widetilde{PR}) & s.t. & \lambda^T A + \beta^T \tilde{C} \geq f \\ & & \lambda, \beta \geq 0 \end{array}$$

Which is equivalent to

$$\min_{\lambda \geq 0} \left\{ \lambda^T b + \min_{\beta \geq 0} \left\{ \beta^T \tilde{d} : \beta^T \tilde{C} \geq f - \lambda^T A \right\} \right\}$$

PUC-Rio - Certificação Digital Nº 0611954/CA

Then, applying linear duality to the inner problem, we obtain

$$\min_{\lambda \geq 0} \left\{ \lambda^T b + \max_{x \geq 0} \left\{ (f - \lambda^T A)x : \tilde{C}x \leq \tilde{d} \right\} \right\}$$

or

$$\min_{\lambda \geq 0} \left\{ \max_{x \geq 0} \left\{ fx + (b - Ax)\lambda^T : \tilde{C}x \leq \tilde{d} \right\} \right\} = v(LD)$$

■

This Theorem deals with the strength of the Lagrangean relaxation and it allows us to draw the following conclusions. The Lagrangean relaxation value, $v(LD)$, is at least as strong as the traditional linear programming relaxation, $v(LP)$. In fact, they have the same value when $Co\{x \in \{0,1\}^n \,|Cx \leq d\} = \left\{x \in \mathbb{R}^n_+ |Cx \leq d,\ 0 \leq x \leq 1\right\}$, i.e., the vertices of the linear relaxation are all integer, as in the case of total unimodular problems. Thus, if one wants to get advantage of the Lagrangean relaxation, they have to keep a subset of constraints $\{x \in \{0,1\}^n \,|Cx \leq d\}$ such that the resulting problem is easy to solve, but at the same time does not have the integrality property. Figure 4.1 summarizes this discussion.

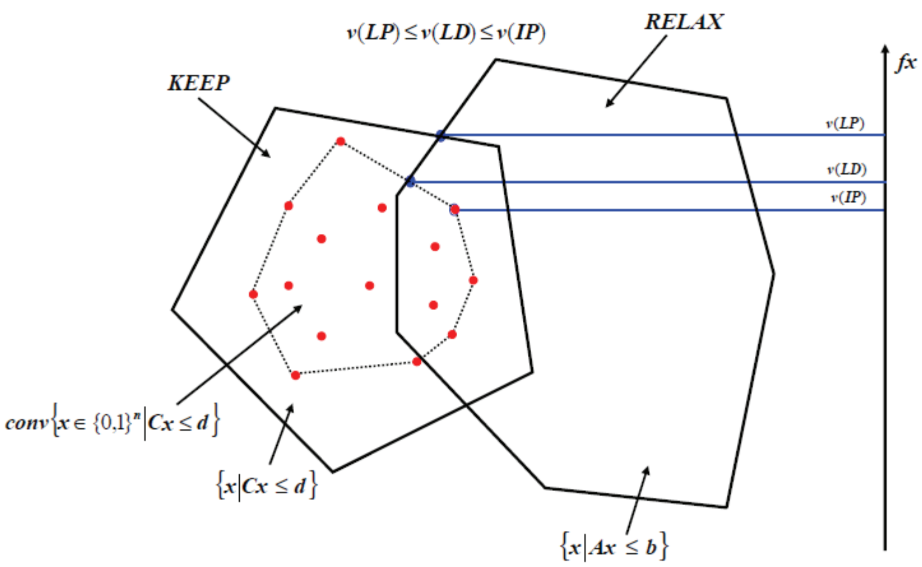

Figure 4.1: Comparison between Relaxations and the (IP) optimum solution $v(IP)$

(a) Lagrangean Decomposition

An interesting extension of the Lagrangean relaxation is the Lagrangean decomposition (see Guignard and Kim [Gui87]). It amounts at rewriting (IP)

PUC-Rio - Certificação Digital Nº 0611954/CA

in its equivalent form, where its variables are duplicated.

$$
\begin{aligned}
& & \max \quad & f^T(x+y) & \\
(IP) & & s.t. \quad & Ax \leq b & (4.23)\\
& & & Cy \leq d & (4.24)\\
& & & x = y & (4.25)\\
& & & x, y \in \{0,1\}^n &
\end{aligned}
$$

Dualizing the constraint $x = y$, the problem decomposes naturally into two Lagrangean relaxation problems:

$$
\begin{array}{lll}
 & \max & (f^T/2+\lambda)x \\
(LR1_\lambda) & s.t. & Ax \geq b \\
 & & x \in \{0,1\}^n
\end{array}
\quad \text{and} \quad
\begin{array}{lll}
 & \max & (f^T/2-\lambda)y \\
(LR2_\lambda) & s.t. & Cy \geq d \\
 & & y \in \{0,1\}^n
\end{array}
$$

Then the Lagrangean dual bound is obtained by solving $\min_\lambda(LR1_\lambda + LR2_\lambda)$.

It is important to notice that when one dualizes equality constraints, the associated lagrangean multipliers are unrestricted in sign. Also, a feasible lagrangean solution, i.e., one such that $x = y$, is automatically optimal to the original integer solution. Additionally, one can easily show using a similar reasoning as in the proof of Theorem 4.2 that the Lagrangean decomposition bounds are at least as strong as the Lagrangean relaxation ones.

PUC-Rio - Certificação Digital Nº 0611954/CA