# Decomposition Methods

## 1.0 Introduction

Consider the XYZ corporation that has 10 departments, each of which have a certain function necessary to the overall productivity of the corporation. The corporation has capability to make 100 different products, but it any particular month, it makes some of them and does not make others. The decision of which products to make is made by the CEO. The CEO's decision is a yes/no decision on each of the 100 products. Of course, the CEO's decision depends, in part, on the productivity of the departments and their capability to make a profit given the decision of which products to make.

Each department knows its own particular business very well, and each has developed sophisticated mathematical programs (optimization problems) which provide the best (most profitable) way to use their resources given identification of which products to make.

The organization works like this. The CEO makes a tentative decision on which products to make, based on his/her own mathematical program which assumes certain profits from each department based on that decision. He/she then passes that decision to the various departments. Each of the departments uses that information to determine how it is going to operate in order to maximize profitability. Then each department passes that information back to the CEO. Some departments may not be able to be profitable at all with the CEO's selection of products to make: these departments will also communicate this information to the CEO.

Once the CEO gets all the information back from the departments, he or she will re-reun their optimization to select the products, likely resulting in a modified choice of products, and then the process will repeat. At some point, the optimization problem solved by the CEO will not change from one iteration to the next. At this point, the CEO will believe the current selection of products is best.

This is an example of a *multidivisional problem* [1, pg. 219]. Such problems involve coordinating the decisions of separate divisions, or departments, of a large organization, when the divisions operate autonomously. Solution of such problems often may be facilitated by separating them into a single master problem and subproblems where the master corresponds to the problem addressed by the CEO and the subproblems correspond to the problems addressed by the various departments.

We developed our example where the master problem involved choice of integer variables and the subproblems involved choice of continuous variables. The reason for this is that it conforms to the form of a mixed-integer-programming (MIP) problem, which is the kind of problem we have most recently had interest.

However, the master-subproblem relationship may be otherwise. It may also involve decisions on the part of the CEO to directly modify resources for each department. By "resources," we mean the right-hand-side of the constraints. Such a scheme is referred to as a *resource-directed* approach.

Alternatively, the master-subproblem relationship may involve decisions on the part of the CEO to indirectly modify resources by charging each department a price for the amount of resources that are used. The CEO would then modify the prices, and the departments would adjust accordingly. Such a scheme is called a *price-directed* approach.

These types of optimization solutions are referred to as <u>decomposition methods</u>.

## 2.0 Connection with optimization: problem structure [2]

Recall that our optimization problems were always like this:

$$\begin{aligned} &\text{Minimize } f(\underline{x}) \\ &\text{Subject to } \ \underline{c}_1\underline{x} \le b_1 \\ &\qquad\qquad\quad \underline{c}_2\underline{x} \le b_2 \\ &\qquad\qquad\quad \ldots \\ &\qquad\qquad\quad \underline{c}_m\underline{x} \le b_m \end{aligned}$$

We may place all of the row-vectors $c_i$ into a matrix, and all elements $b_i$ into a vector $\underline{b}$, so that our optimization problem is now:

$$\begin{aligned} &\text{Minimize } f(\underline{x}) \\ &\text{Subject to } \ \underline{A}\ \underline{x} \le \underline{b} \end{aligned}$$

Problems that have special structures in the constraint matrices $\underline{A}$ are typically more amenable to decomposition methods. Almost all of these structures involve the constraint matrix $\underline{A}$ being block-angular. A block angular constraint matrix is illustrated in Fig. 1. In this matrix, the yellow-cross-hatched regions represent sub-matrices that contain non-zero elements. The remaining sub-matrices, not yellow-cross-hatched, contain all zeros. We may think of each yellow-cross-hatched as a department. The decision variables $\underline{x}_1$ are important only to department 1; the decision variables $\underline{x}_2$ are important only to department 2; and the decision variables $\underline{x}_3$ are important only to department 3. In this particular structure, we have no need of a CEO at all. All departments are completely independent!

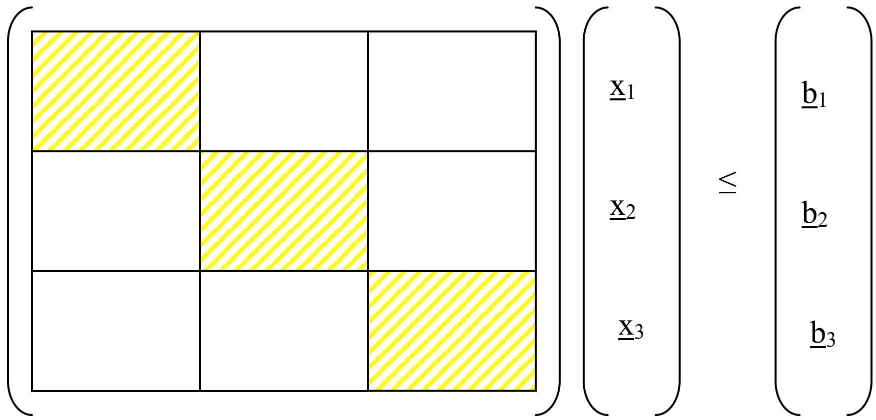


Fig. 1: Block-angular structure

Fig. 2 represents a structure where the decisions are linked at the CEO level, who must watch out for the entire organization's consumption of resources. In this case, the departments are independent, i.e., they are concerned only with decision on variables for which no other department is concerned, BUT… the CEO is concerned with constraints that span across the variables for all departments. And so we refer to this structure as block-angular with linking constraints.

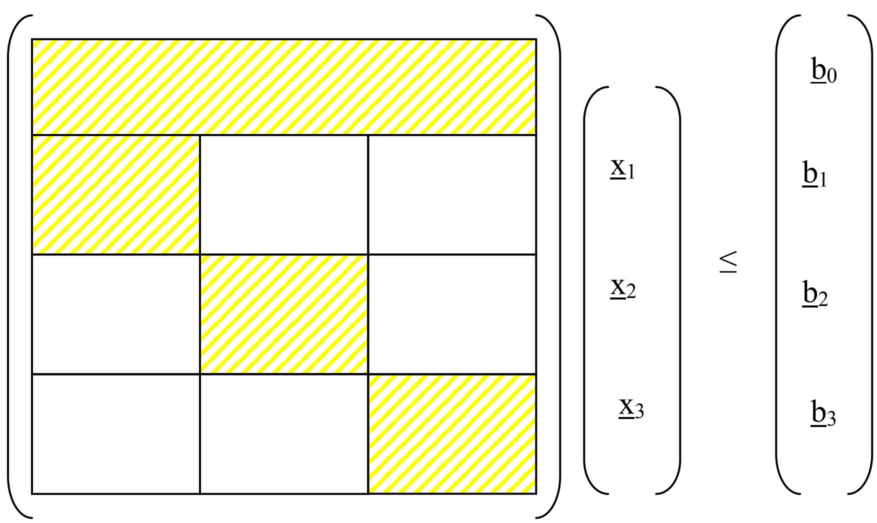


Fig. 2: Block-angular structure with linking constraints

This previous situation, characterized by Fig. 2, is not, however, the situation that we originally described. In the situation of Fig. 2, the CEO allocates resources.

In the original description, the CEO chose values (1 or 0) for certain variables for which it was assumed would affect each department. The original situation would have different departments linked by variables, then, and not by constraints. The structure of the constraint matrix for this situation is shown in Fig. 3.

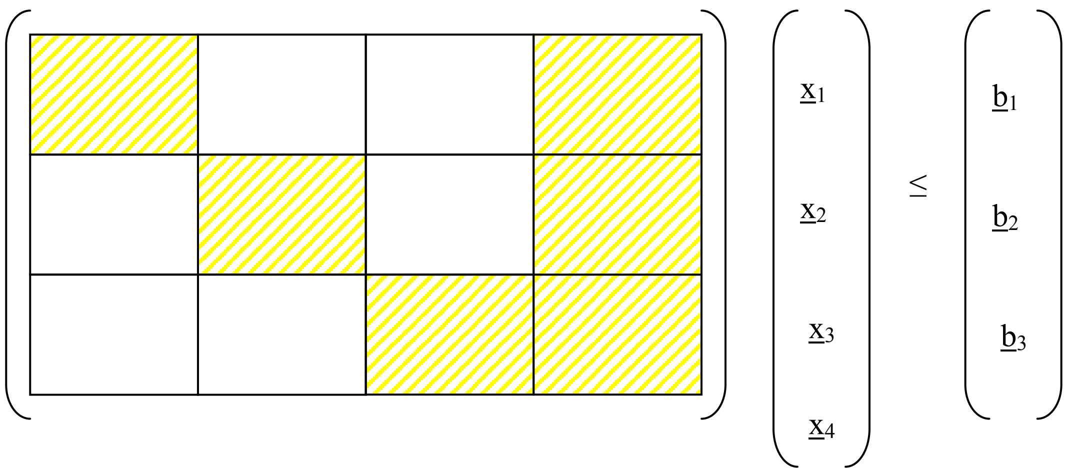


Fig. 3: Block-angular structure with linking variables

## 3.0 Motivation for decomposition methods: solution speed

To motivate decomposition methods, we consider introducing security constraints to a familiar problem: the OPF.

The OPF may be posed as problem $P_0$.

$$
P_0 \qquad \begin{aligned} &Min && f_0(x_0, u_0) && \\ &s.t. && h_k(x_k, u_0) = 0 && k = 0 \\ & && g_k(x_k, u_0) \le g_k^{\max} && k = 0 \end{aligned}
$$

where $h_k(x_k,u_0)=0$ represents the power flow equations and $g_k(x_k,u_0)\leq g_k^{max}$ represents the line-flow constraints. The index k=0 indicates this problem is posed for only the "normal condition," i.e., the condition with no contingencies.

Denote the number of constraints for this problem as N.

Assumption: Let's assume that running time T of the algorithm we use to solve the above problem is proportional to the square of the number of constraints, i.e., $N^2$. For simplicity, we assume the constant of proportionality is 1, so that $T=N^2$.

Now let's consider the security-constrained OPF (SCOPF). Its problem statement is given as problem $P_c$:

$$P_c \qquad \begin{array}{lll} Min & f_0(x_0,u_0) & \\ s.t. & h_k(x_k,u_0)=0 & k=0,1,2,...,c \\ & g_k(x_k,u_0)\leq g_k^{max} & k=0,1,2,...,c \end{array}$$

Notice that there are *c* contingencies to be addressed in the SCOPF, and that there are a complete new set of constraints for each of these *c* contingencies. Each set of contingency-related constraints is exactly like the original set of constraints (those for problem $P_0$), except it corresponds to the system with an element removed.

So the SCOPF must deal with the original N constraints, and also another set of N constraints for every contingency. Therefore, the total number of constraints for Problem $P_C$ is N+cN=(c+1)N.

Under our assumption that running time is proportional to the square of the number of constraints, then the running time will be proportional to $[(c+1)N]^2=(c+1)^2N^2=(c+1)^2T$.

What does this mean?

It means that the running time of the SCOPF is $(c+1)^2$ times the running time of the OPF. So if it takes OPF 1 minute to run, and you want to run SCOPF with 100 contingencies, it will take you $101^2$ minutes, or 10,201 minutes to run the SCOPF. This is 170 hours, about 1 week!!!!

Many systems need to address 1000 contingencies. This would take about 2 years!

So this is what you do…..

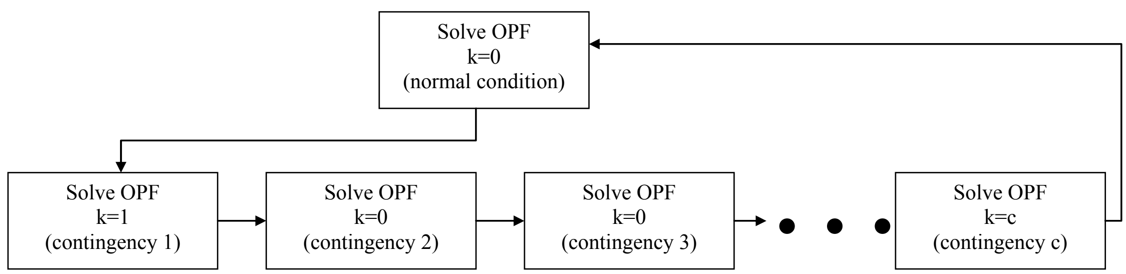


Fig. 4: Decomposition solution strategy

The solution strategy first solves the OPF (master problem) and then takes contingency 1 and re-solves the OPF, then contingency 2 and resolves the OPF, and so on (these are the subproblems). For any contingency-OPFs which require a redispatch, relative to the k=0 OPF, an appropriate constraint is generated, at the end of the cycle, these constraints are gathered and applied to the k=0 OPF. Then the k=0 OPF is resolved, and the cycle starts again. Experience has it that such an approach usually requires only 2-3 cycles.

Denote the number of cycles as $m$.

Each of the individual problems has only N constraints and therefore requires only T minutes.

There are (c+1) individual problems for every cycle.

There are m cycles.

So the amount of running time is m(c+1)T.

If c=100 and m=3, T=1 minute, this approach requires 303 minutes. That would be about 5 hours (instead of 1 week).

If c=1000 and m=3, T=1 minute, this approach requires about 50 hours (instead of 2 years).

In addition, this approach is easily parallelizable, i.e., each individual OPF problem can be sent to its own CPU. This will save even more time.

Fig. 5 compares computing time for a "toy" system. The comparison is between a full SCOPF, a decomposed SCOPF (DCOPF), and a decomposed SCOPF where the individual OPF problems have been sent to separate CPUs.

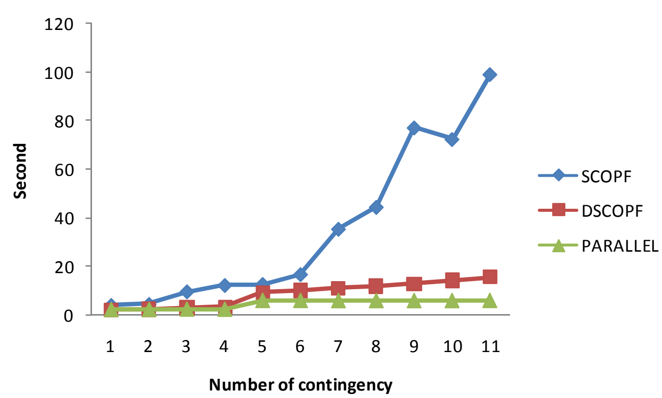


Fig. 5

## 4.0 Benders decomposition

J. F. Benders [3] proposed solving a mixed-integer programming problem by partitioning the problem into two parts – an integer part and a continuous part. It uses the branch-and-bound method on the integer part and linear programming on the continuous part.

The approach is well-characterized by the linking-variable problem illustrated in Fig. 3 where here the linking variables are the integer variables. We will introduce this algorithm in this section using an example.

Consider the following problem $P_0$ [4]:

$$
P_0 \qquad
\begin{aligned}
\max \quad & z_1 = 4x_1 + 3x_2 + 5w \\
\text{subject to:} \quad & 2x_1 + 3x_2 + w \le 12 \\
& 2x_1 + x_2 + 3w \le 12 \\
& w \le 20 \\
& x_1, x_2, w \ge 0, \\
& w \text{ integer}
\end{aligned}
$$

Clearly it is a mixed integer problem.

Let's redefine the objective function as

$$z_1 = 5w + z_2$$

where

$$z_2 = 4x_1 + 3x_2$$

so that problem $P_1$ below is equivalent to problem $P_0$ above:

$$
\mathrm{P}_{1\mathrm{P}} \quad
\begin{aligned}
&\max && z_1 = 5w + \left(\begin{aligned} &\max && z_2 = 4x_1 + 3x_2 \\ &\text{subject to} && 2x_1 + 3x_2 \le 12 - w \\ &&& 2x_1 + x_2 \le 12 - 3w \\ &&& x_1, x_2 \ge 0 \end{aligned}\right) \\
&\text{subject to:} && w \le 20 \\
&&& w \ge 0, \\
&&& w \text{ integer}
\end{aligned}
$$

We can think of the problem inside the brackets of $P_1$ as a stand-alone problem where $w$ is specified. Call this problem $P_{11}$:

$$
\mathrm{P}_{11} \quad
\begin{aligned}
&\max && z_2 = 4x_1 + 3x_2 \\
&\text{subject to} && 2x_1 + 3x_2 \le 12 - w \\
&&& 2x_1 + x_2 \le 12 - 3w \\
&&& x_1, x_2 \ge 0
\end{aligned}
$$

Assume the solution to $P_{11}$ is $z_2$*. Then $P_{1P}$ may be written as $P_1$*

$$
\mathrm{P}_1^* \quad
\begin{aligned}
&\max && z_1 = 5w + \left(z_2^*\right) \\
&\text{subject to:} && w \le 20 \\
&&& z_2^* \le M \\
&&& w \ge 0, \\
&&& w \text{ integer}
\end{aligned}
$$

Notice that we added the constraint $z_2^* \le M$ in $P_1$* in order to bound $z_2$*. Here, we assume that M is just some arbitrarily large number.

Recall the duality theorem, which says that the optimum value of the dual of an LP is the same as the optimum value of the LP. Therefore the optimum value of the dual of $P_{11}$, is the optimum value of $P_{11}$, which is $z_2$*. Let's write down the dual of $P_{11}$. It is

$$
D_{11} \quad
\begin{aligned}
&\min && z_2 = (12-w)\lambda_1 + (12-3w)\lambda_2 \\
&\text{subject to} && 2\lambda_1 + 2\lambda_2 \geq 4 \\
& && 3\lambda_1 + \lambda_2 \geq 3 \\
& && \lambda_1, \lambda_2 \geq 0
\end{aligned}
$$

In both $P_{11}$ and $D_{11}$, $w$ is specified (it is not a variable).

Since $P_{11}$ and $D_{11}$ give the same solution at the optimum, we can replace the maximization problem in $P_1$ with the dual problem $D_{11}$.

$$
P_{1D} \quad
\begin{aligned}
&\max && z_1 = 5w + \left(
\begin{aligned}
&\min && z_2 = (12-w)\lambda_1 + (12-3w)\lambda_2 \\
&\text{subject to} && 2\lambda_1 + 2\lambda_2 \geq 4 \\
& && 3\lambda_1 + \lambda_2 \geq 3 \\
& && \lambda_1, \lambda_2 \geq 0
\end{aligned}
\right) \\
&\text{subject to:} && w \leq 20 \\
& && w \geq 0, \\
& && w \text{ integer}
\end{aligned}
$$

Recalling our notation that the optimum value of the dual problem $D_{11}$, which is inside the brackets of $P_{1D}$, is $z_2$*, we can rewrite $P_{1D}$ just as we rewrote $P_{1P}$, as $P_1$*:

$$
P_1^* \quad
\begin{aligned}
&\max && z_1 = 5w + \left(z_2^*\right) \\
&\text{subject to:} && w \leq 20 \\
& && z_2^* \leq M \\
& && w \geq 0, \\
& && w \text{ integer}
\end{aligned}
$$

To solve $P_1$*, normally, we would use Branch & Bound, but this is a very simple problem, and one can see that the solution is *w*=20*. Now let's use this solution, *w*=20*, in $D_{11}$; we will call the resulting problem $D_{11a}$:

$$
D_{11a} \qquad \begin{array}{ll} \min & z_2 = (12-20)\lambda_1 + (12-3*20)\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}
$$

which is

$$
D_{11a} \qquad \begin{array}{ll} \min & z_2 = -8\lambda_1 - 48\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}
$$

We would normally solve $D_{11a}$ as an LP, but in this case, it is easy to see that $D_{11a}$ is unbounded since the constraints bound $\lambda_1$ and $\lambda_2$ from below, and the objective function minimizes negated values of $\lambda_1$ and $\lambda_2$. So we can make $\lambda_1$ and $\lambda_2$ as large as we like, making $z_2$ smaller and smaller (more and more negative).

Now what does an unbounded $D_{11}$ problem mean? Referring back to our notes on duality, we see that if the dual is unbounded, the primal must be infeasible.

This means that $P_{11}$ (the primal) must be infeasible. Therefore, when $w$=20, one or more constraints in $P_{11}$ are violated for *any* values of $x_1$ and $x_2$. This means our choice of $w$=20 must not be a good one.

Now recall where we obtained $w=20$. We obtained it from solving $P_1$*, which was a relaxed version of $P_0$. Since the constraints of the primal $P_{11}$ are also constraints of the original mixed integer problem $P_0$, it must be the case that we need to add constraints to $P_1$* to restrict what $w$ can be.

We can think either in terms of making the primal $P_{11}$ feasible, or we can think in terms of making the dual bounded. Let's do the latter. To make the dual bounded, refer back to $D_{11}$, and ask the following question:

How to choose *w* to make this problem bounded?

The answer to this question must lie in the objective function of $D_{11}$ because that is the only place that *w* appears. So our question is:

How to choose *w* in the objective function to make $D_{11}$ bounded?

$D_{11}$ is unbounded because as we increase $\lambda_1$ and $\lambda_2$, the objective function gets smaller (more and more negative).

But this would not be the case if the coefficients of $\lambda_1$ and $\lambda_2$ were positive! $D_{11}$ is repeated below to facilitate this observation:

$$
D_{11} \quad
\begin{aligned}
\min \quad & z_2 = (12-w)\lambda_1 + (12-3w)\lambda_2 \\
\text{subject to} \quad & 2\lambda_1 + 2\lambda_2 \geq 4 \\
& 3\lambda_1 + \lambda_2 \geq 3 \\
& \lambda_1, \lambda_2 \geq 0
\end{aligned}
$$

To force the coefficients of $\lambda_1$ and $\lambda_2$ to be positive, we require:

$$(12-w) \geq 0 \Rightarrow w \leq 12$$
$$(12-3w) \geq 0 \Rightarrow w \leq 4$$

Clearly the second constraint is more restrictive. This *feasibility cut* is the one that we will impose on $P_1$*, resulting in $P_2$*:

$$
P_2^* \quad
\begin{aligned}
\max \quad & z_1 = 5w + (z_2^*) \\
\text{subject to:} \quad & w \leq 20 \\
& w \leq 4 \\
& z_2^* \leq M \\
& w \geq 0, \\
& w \text{ integer}
\end{aligned}
$$

The solution to $P_2$* is clearly *w=4*, $z_2$*=M, with $z_1$*=5(4)+M=20+M.. Substituting this solution into $D_{11}$ results in $D_{11b}$

$$D_{11b} \quad \begin{array}{ll} \min & z_2 = (12-4)\lambda_1 + (12-3*4)\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

which is:

$$D_{11b} \quad \begin{array}{ll} \min & z_2 = 8\lambda_1 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

Using CPLEX LP solver, we find the solution to $D_{11b}$ is $\lambda_1$=0, $\lambda_2$=3, with objective function value $z_2$*=0. (Clearly this problem requires $\lambda_1$=0; then $\lambda_2$ can be anything as long as it satisfies

$$\begin{aligned} &2\lambda_2 \geq 4 \Rightarrow \lambda_2 \geq 2 \\ &\lambda_2 \geq 3 \end{aligned}$$

Therefore a finite optimal solution is $\lambda_1$=0, $\lambda_2$=3.)

Using $z_2$*=0 in our overall objective function, $z_1 = 5w + (z_2^* \quad)$, (with *w=4*) we obtain $z_1$=20+($z_2$*)=20+0=20. Yet, our last solution to $P_2$* resulted in $z_1$=20+M.

What this tells us is that the real objective function value at the optimal must be between 20 and 20+M.

The reason we obtained 20+M in $P_2$* must be because the constraint $z_2$*≤M was not restrictive enough. Recall $D_{11}$:

$$\text{D}_{11} \qquad \begin{array}{ll} \min & z_2 = (12-w)\lambda_1 + (12-3w)\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

We can use the solution just obtained, $\lambda_1=0$, $\lambda_2=3$, to impose a new constraint on $z_2$*, according to:

$$z_2 = (12-w)(0) + (12-3w)(3) = 36 - 9w$$

What we know is that the solution obtained in $D_{11b}$ was too large, so that

$$z_2 \leq 36 - 9w$$

This first *optimality cut* is the one that we will impose on $P_2$*, resulting in $P_3$*:

$$\text{P}_3^* \qquad \begin{array}{ll} \max & z_1 = 5w + \left(\text{z}_2^* \quad\right) \\ \text{subject to:} & w \leq 20 \\ & w \leq 4 \\ & \text{z}_2^* \leq \text{M} \\ & z_2 \leq 36 - 9w \\ & w \geq 0, \\ & w \text{ integer} \end{array}$$

Now we could use Branch and Bound on this all-integer problem, but you can see the answer by enumeration for *w=0*, *w=1*, *w=2*, *w=3*, *w=4*. We see that *w=0*, $z_2^*=36$, $z_1^*=36$ is the answer.

Now we have a new solution: *w=0*. Substitution of this into $D_{11}$ results in $D_{11c}$:

$$\text{D}_{11c} \qquad \begin{array}{ll} \min & z_2 = 12\lambda_1 + 12\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

Using CPLEX LP solver, we find the solution to $D_{11c}$ is $\lambda_1=2$, $\lambda_2=0$, with objective $z_2^*=24$.

Using this $z_2^*$ in our overall objective function, $z_1 = 5w + (z_2^*)$, (with $w=0$), we obtain $z_1=0+(z_2^*)=0+24=24$. Yet, our last solution to $P_2^*$ resulted in $z_1=0+36$.

What this tells us is that the real objective function value at the optimal must be between 24 and 36.

The reason we obtained 36 in $P_3^*$ must be because the constraints $z_2^* \leq M$ and $z_2 \leq 36-9w$ were not restrictive enough. Recall $D_{11}$:

$$
D_{11} \qquad
\begin{aligned}
\min \quad & z_2 = (12-w)\lambda_1 + (12-3w)\lambda_2 \\
\text{subject to} \quad & 2\lambda_1 + 2\lambda_2 \geq 4 \\
& 3\lambda_1 + \lambda_2 \geq 3 \\
& \lambda_1, \lambda_2 \geq 0
\end{aligned}
$$

We can use the solution just obtained, $\lambda_1=2$, $\lambda_2=0$, to impose a new constraint on $z_2^*$, according to:

$$z_2 = (12-w)(2) + (12-3w)(0) = 24-2w$$

What we know is that the solution obtained in $D_{11c}$ was too large, so

$$z_2 \leq 24-2w$$

This second *optimality cut* is the one that we will impose on $P_3^*$, resulting in $P_4^*$:

$$
P_4^* \qquad
\begin{aligned}
\max \quad & z_1 = 5w + \left(z_2^*\right) \\
\text{subject to:} \quad & w \le 20 \\
& w \le 4 \\
& z_2^* \le M \\
& z_2 \le 36 - 9w \\
& z_2 \le 24 - 2w \\
& w \ge 0, \\
& w \text{ integer}
\end{aligned}
$$

Now we could use Branch and Bound on this all-integer problem, but you can see the answer by enumeration for *w=0*, *w=1*, *w=2*, *w=3*, *w=4*. We see that $w=2$, $z_2^*=18$, $z_1^*=28$ is the answer.

Now we have a new solution: $w=2$. Substitution of this into $D_{11}$ results in $D_{11d}$:

$$
D_{11d} \qquad
\begin{aligned}
\min \quad & z_2 = 10\lambda_1 + 6\lambda_2 \\
\text{subject to} \quad & 2\lambda_1 + 2\lambda_2 \ge 4 \\
& 3\lambda_1 + \lambda_2 \ge 3 \\
& \lambda_1, \lambda_2 \ge 0
\end{aligned}
$$

Using CPLEX LP solver, we find the solution to $D_{11d}$ is $\lambda_1=0.5$, $\lambda_2=1.5$, with objective function value $z_2^*=14$.

Using this $z_2^*$ in our overall objective function, $z_1 = 5w + \left(z_2^*\right)$, (with $w=2$), we obtain $z_1=5(2)+(z_2^*)=10+14=24$. Yet, our last solution to $P_4^*$ resulted in $z_1=0+28$.

What this tells us is that the real objective function value at the optimal must be between 24 and 28.

The reason we obtained 28 in $P_4^*$ must be because the constraints $z_2^* \le M$, $z_2 \le 36 - 9w$, and $z_2 \le 24 - 2w$ were not restrictive enough. Recall $D_{11}$:

$$D_{11}\quad \begin{aligned} \min \quad & z_2 = (12-w)\lambda_1 + (12-3w)\lambda_2 \\ \text{subject to} \quad & 2\lambda_1 + 2\lambda_2 \ge 4 \\ & 3\lambda_1 + \lambda_2 \ge 3 \\ & \lambda_1, \lambda_2 \ge 0 \end{aligned}$$

We can use the solution just obtained, $\lambda_1$=0.5, $\lambda_2$=1.5, to impose a new constraint on $z_2$*, according to:

$$z_2 = (12-w)(0.5) + (12-3w)(1.5) = 6 - 0.5w + 18 - 4.5w = 24 - 5w$$

What we know is that the solution obtained in $D_{11d}$ was too large, so

$$z_2 \le 24 - 5w$$

This third *optimality cut* is the one that we will impose on $P_4$*, resulting in $P_5$*:

$$P_5^* \quad \begin{aligned} \max \quad & z_1 = 5w + \left(z_2^*\right) \\ \text{subject to:} \quad & w \le 20 \\ & w \le 4 \\ & z_2^* \le M \\ & z_2 \le 36 - 9w \\ & z_2 \le 24 - 2w \\ & z_2 \le 24 - 5w \\ & w \ge 0, \\ & w \text{ integer} \end{aligned}$$

Now enumerating the solutions to this problem results in

*w=0*: $z_2$*=24, $z_1$*=24

*w=1*: $z_2$*=19, $z_1$*=24

*w=2*: $z_2$*=14, $z_1$*=24

*w=3*: $z_2$*=9, $z_1$*=24

*w=4*: $z_2$*=0, $z_1$*=20

We see that *w*=0, 1, 2, and 3 are equally good solutions

For each of these solutions, let's compute the dual. Substution of each one into $D_{11}$,

$$D_{11} \quad \begin{array}{ll} \min & z_2 = (12-w)\lambda_1 + (12-3w)\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

results in the following:

<u>*w=0*</u>:

$$D_{11e\text{-}1} \quad \begin{array}{ll} \min & z_2 = 12\lambda_1 + 12\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

Solution is $\lambda_1$=2, $\lambda_2$=0, with objective function value $z_2$*=24. Using this $z_2$* in our overall objective function, $z_1 = 5w + (z_2^*\ )$, (with $w$=0), we obtain $z_1$=5(0)+($z_2$*)=0+24=24. Our last solution to $P_5$* resulted in $z_1$=24. This solution is optimal. Dual variables obtained from CPLEX are $x_1$=6, $x_2$=0.

<u>*w=1*</u>:

$$D_{11e\text{-}2} \quad \begin{array}{ll} \min & z_2 = 11\lambda_1 + 9\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

Solution is $\lambda_1$=0.5, $\lambda_2$=1.5, with objective function value $z_2$*=19. Using this $z_2$* in our overall objective function, $z_1 = 5w + (z_2^*\ )$, (with $w$=1), we obtain $z_1$=5(1)+($z_2$*)=5+19=24. Our last solution to $P_5$* resulted in $z_1$=24. This solution is optimal. Dual variables obtained from CPLEX are $x_1$=4, $x_2$=1.

$\underline{w=2}$:

$$D_{11e\text{-}3}\quad \begin{array}{ll} \min & z_2 = 10\lambda_1 + 6\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

Solution is $\lambda_1$=0.5, $\lambda_2$=1.5, with objective function value $z_2$*=14. Using this $z_2$* in our overall objective function, $z_1 = 5w + \left(z_2^*\right)$, (with $w$=2), we obtain $z_1$=5(2)+($z_2$*)=10+14=24. Our last solution to $P_5$* resulted in $z_1$=24. This solution is optimal. Dual variables obtained from CPLEX are $x_1$=2, $x_2$=2.

$\underline{w=3}$:

$$D_{11e\text{-}4}\quad \begin{array}{ll} \min & z_2 = 9\lambda_1 + 3\lambda_2 \\ \text{subject to} & 2\lambda_1 + 2\lambda_2 \geq 4 \\ & 3\lambda_1 + \lambda_2 \geq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{array}$$

Solution is $\lambda_1$=0, $\lambda_2$=3, with objective function value $z_2$*=9. Using this $z_2$* in our overall objective function, $z_1 = 5w + \left(z_2^*\right)$, (with $w$=3), we obtain $z_1$=5(3)+($z_2$*)=15+9=24. Our last solution to $P_5$* resulted in $z_1$=24. This solution is optimal. Dual variables obtained from CPLEX are $x_1$=0, $x_2$=3.

In summary, recall our original problem:

$$P_0\quad \begin{array}{ll} \max & z_1 = 4x_1 + 3x_2 + 5w \\ \text{subject to:} & 2x_1 + 3x_2 + w \leq 12 \\ & 2x_1 + x_2 + 3w \leq 12 \\ & w \leq 20 \\ & x_1, x_2, w \geq 0, \\ & w \text{ integer} \end{array}$$

Optimal solutions to this problem result in an objective function value of $z_1$=24 and are:

- $w$=0, $x_1$=6, $x_2$=0
- $w$=1, $x_1$=4, $x_2$=1
- $w$=2, $x_1$=2, $x_2$=2
- $w$=3, $x_1$=0, $x_2$=3

It is coincidence that the values of $x_1$ and $x_2$ for the optimal solution also turn out to be integers.

The fact that there are multiple solutions is typical of MIP problems. MIP problems are generally non-convex.

## 5.0 Benders decomposition – a summary

There are basically three steps to Benders decomposition for mixed integer programs.

The problem can be generally specified as follows:

$$
\begin{aligned}
&\max z_1 = c_1^T x + c_2^T w \\
&s.t. \\
&A_1 x + A_2 w \le b \\
&x, w \ge 0 \\
&w \quad \text{integer}
\end{aligned}
$$

where we specify:

Master problem:

$$\max z_1 = c_2^T w + z_2^*$$
$$s.t.$$
$$w \text{ constrained by constraints in } Q$$
$$z_2^* \le M$$
$$w \quad \text{integer}$$

Sub-problem (dual):

$$\min z_2 = (b - A_2 w)^T \lambda$$
$$s.t.$$
$$w \text{ constrained by constraints in } Q$$
$$A_1^T \lambda \ge c_1$$
$$\lambda \ge 0$$

We begin by defining Q as the set of constraints binding the master (all-integer) program. Initially, Q={w≤large number}.

1. Solve the following problem using Branch and Bound (or any other integer programming method). Designate the solution as w*.
2. Using the value of w* found in step 1, solve the sub-problem. If the solution is unbounded, adjoin a feasibility constraint to Q and go to step 1. Otherwise, designate the solution as λ* and go to step 3.
3. If the value of $z_1$ found in step 1 exactly equals $c_2$w*+$z_2$, where $z_2$ is found in step 2, the solution w*, x* corresponding to the dual of the subproblem solution, is optimal. Otherwise, adjoin an optimality constraint to Q and go to step 1.

[1] F. Hillier and G. Lieberman, “Introduction to Operations Research,” 4th edition, Holden-Day, Oakland California, 1986.
[2] M. Minoux, “Mathematical Programming: Theory and Algorithms,” Wiley, 1986.
[3] J. Benders, “Partitioning procedures for solving mixed variables programming problems,” Numerische Mathematics, 4, 238-252, 1962.
[4] S. Zionts, “Linear and Integer Programming,” Prentice-Hall, 1974.