**SOLUTION OF GDP PROBLEMS USING THE D-SDA**

**Notation**

Matrix: Capital and bold

Vector: Lower case and bold

Set: Capital

1. **Generalizing the problem**

Suppose that we have a general GDP of the form:

To apply the reformulation, we must make a distinction between those Boolean and integer variables/Boolean variables that are reformulated using the external variables’ reformulation and the trivial reformulation

.

For Boolean variables in , the first requirement to apply the external variables reformulation is that those must be defined in ordered sets. In this case we will consider a total of ordered sets:

In this case represents the first element of , which is defined by the user. is the cardinality (size) of . This is required because the ordered set does not necessarily start at 1, or worse, the set defined by the user does not have numerical values, e.g {A,B,C,D,E,…}. We make a distinction between and : the first considers the sets over the Boolean variables are defined in the optimization problem, and the second uses subsets , where the reformulation is going to be applied. In other words, is retrieved form variable definition and is retrieved from the “Exactly”-type constraint explained below.

The second requirement is that exactly Boolean variables are expected to be True over a subset :

**\***Important note: for the moment we will not consider the case where there are multiple () “Exactly” constraints with different subsets for a single vector of variables (future work). Another aspect left for future work is the fact that we can increase the number of reformulation layers to include reformulations for constraints of the form “atleast” and “atmost” (future work).

The last requirement is that each vector must represent either discretized locations, discretized points in time or an integer number.

Also, the vector must be identified by the user (in principle any integer or Boolean variable can be assigned to , but that depends on the requirements of the user). Those Boolean variables that are going to be transformed into external variables using the trivial transformation can be included in this vector.

It is important to retrieve from the model the upper and lower bound of these integer variables, to consider those parameters un the upper optimization layer:

At this point, we can count the number of external variables to be defined: . The external variables are:

1. **Generalizing the reformulation**

means: Boolean variable at position of set . Subscript means that it can be reformulated using external variables. Note that the reformulation is only applied to variables defined over , where the “Exactly” constraint was imposed.

Note that for the case with , we have:

This means that the constraint in blue must not be used when . So far, we have not applied the D-SDA to cases with . An example of this is the catalytic distillation problem, where exactly 3 reactive trays must be allocated.

1. **Generalizing the decomposition**

Boolean variables and integer variables are going to be reformulated with external variables. If we apply the reformulation, we obtain an upper and a lower optimization layer. In the upper layer:

In the lower layer:

Note that some disjuncts will be fixed in the subproblems because either is fixed or allows to fix other disjuncts.

**\*\*\***Final comment: In this case I did not consider the case where there are also binary variables defined over ordered sets that can be reformulated into external variables. Are we going to consider this case, i.e. our variables are and? If that is the case, binary variables can be included in this mathematical framework/notation.

**AUTOMATIZING THE REFORMULATION: REQUIREMENTS, STEPS AND EXAMPLES**

**Requirements from the user**

The user is required to define the variables that are going to be reformulated.

1. **Identify variables and sets:** The user must identify the integer variables and Boolean variables that are going to be reformulated using the trivial reformulation () and the Boolean variables that can be potentially reformulated using external variables over sets (). For Boolean variables ( the user must also identify the set/sets () where the reformulation occurs if the variable is defined over multiple stets. At this point, we know the number of vectors of external variables we will have in our problem (), although we still do not know the exact number of external variables. Thus, the user/computer must assign elements from and to **,**e.g.:

is defined for ,is defined for ,…, is defined for ; is defined for .

In summary, user tasks are:

1. Identify and
2. For the user must identify
3. Assign elements from and to (Although this can be done automatically by following the order the user uses to input the information).
4. **Specify the Neighborhood:** The user must specify the neighborhood that is going to be used, e.g. , .

**Automatic Reformulation**

With the information provided by the user, it is now possible to automatize the reformulation. I recommend to follow these steps:

1. **Check the information provided by the user for :** Verify that the variables included in are integer (including binary) or Boolean. If not, abort: “Variable X cannot be reformulated using the trivial reformulation”.

Also check that the upper and lower bounds for are declared if they are not binary or Boolean. If not, there should be a warning: “the problem may be unbounded”; however, the solver can still try to solve the problem (or what do you think?).

1. **Identify the variables from that can be reformulated ():** To do so:
2. First check that for every set for there exist exactly one constraint of the form:

Where we define (The elements are retrieved form the formulation of the problem). Thus, assuming that we are working with ordered information (.). If the information provided by the user is not in order, it must be ordered first, i.e. we must always ensure that we are working with an ordered version of .

1. Second, check that .

* If both a. and b. are met for , then with must be included in . The remaining variables in must be moved to .
* If (a. is not satisfied for ) or (If a. is satisfied but b. is not satisfied for ), abort: “The set of variables XX cannot be reformulated with external variables because………”

1. **Verify if the problem can be solved**: The Boolean variables that appear in must not be in and vice versa. If not, an error message must be shown: Abort: “One or more Boolean variables are being reformulated with both the trivial and the external variables reformulation”.
2. **Count the number of external variables to be defined:** The total number of external variables is .
3. **Declare the external variables of the problem:** Now, we can declare the external variables of the problem and specify how each external variable is associated to a variable or a set of variables of the problem.

For each set we declare external variables, and those external variables are used to specify the value of some Boolean variables:

And for each variable in we specify one external variable:

1. **Declare the inequality constraints for the external variables of the problem:** With the information we have collected, we can write the inequality constraints for the upper-layer problem.

**Example 1: Batch Processing**

**Requirements from the user**

1. **Identify variables and sets:**

In this case, the user must indicate that the reformulation with external variables is going to be applied to and with respect to set , for all , i.e. ,, …, , . The vector of Boolean variables that can be potentially reformulated would look like this (The user must not write this vector, this is just for clarification):

In this case, there are not integer variables that can be directly reformulated into external variables. In this example, the vector that contains our external variables has the following form:

Where is defined for **,** for , …,for . for ,…, for **.**

In summary, is assigned to , is assigned to ,…, is assigned to , etc.

1. **Specify the Neighborhood:**

**Automatic Reformulation**

1. **Check the information provided by the user for :**

In this case there is no information for

1. **Identify the variables from that can be reformulated ():**

From **Requirements from the user (1. Identify variables and sets)**, we have .

For , i.e. , we have:

And for ,i.e. ,we have:

Thus, .

1. **Verify if the problem can be solved**:

kkk

1. **Count the number of external variables to be defined:**

kkk

1. **Declare the external variables of the problem:**

kkk

1. **Declare the inequality constraints for the external variables of the problem:**

kkk

**Example 2: Catalytic Distillation (GAMS code DISTILLATION.gms)**

**Requirements from the user**

1. **Identify variables and sets:**

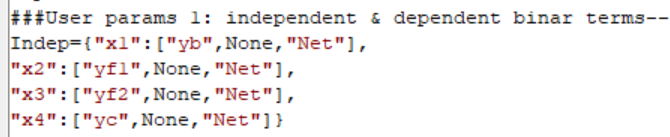
In this case, the user must indicate that the reformulation with external variables is going to be applied to and with respect to set , i.e., , etc. In this case, there are not integer variables that can be directly reformulated into external variables.

The vector of Boolean variables that can be reformulated (potentially):

Thus, the vector that contains our external variables has the following form:

Where is defined for ,for , etc.

In the gams code, this is shown from line 1248 to line 1242:



Where refers to , etc. refers to . This means that is being assigned to , etc. Note that including the external variables as inputs is not mandatory. For example, external variables may me assigned depending on the order used to input the information.

In the GAMS code the user is also required to specify those terms that can be directly calculated from the reformulated terms (lines 1244 and 1245), however, the idea is to avoid this in the Pyomo/Python generalization (am I correct?).

1. **Specify the Neighborhood:** .

**Automatic Reformulation**

1. **Check the information provided by the user for :**

In this case there is no information for

1. **Identify the variables from that can be reformulated ():**

kkk

1. **Verify if the problem can be solved**:

kkk

1. **Count the number of external variables to be defined:**

kkk

1. **Declare the external variables of the problem:**

kkk

1. **Declare the inequality constraints for the external variables of the problem:**

kkk

**\*\*\*Note on the GAMS code: It is not completely general!!!!!!!. What I noticed:**

* It does not consider the reformulation of integer variables or Boolean variables (**)** directly, using the trivial transformation.
* It requires numerical ordered sets of the form , and the first element must be 1.
* It cannot handle the reformulation of variables defined over two indexes, e.g. .
* It requires the problem to be expressed as a minimization problem (Not suitable for maximization)
* Things that are going to be reformulated must be expressed as parameters. Also, parameters that depend on the external variables must be identified by the user.
* It can only handle NLP subproblems, although this may be easily changed when the model is stated.
* Reformulations are applied over the whole . Thus, we are not considering the case where some variables of are not fixed by the external variables.

**GENERALIZING THE ALGORITHM**

A general algorithm would be similar to the scheme we proposed in the rate-based article.

…