PARTIAL RELAXATION OF EQUALITY-CONSTRAINED PROGRAMS

Isaac Siwale

Technical Report RD-24-2015
Apex Research Limited
London
England

e-mail: ike_siwale@hotmail.com

Abstract

This paper presents a reformulation that is a natural “by-product” of the ‘variable endogenization’ process for equality-constrained programs.[1] The method results in a partial relaxation of the constraints which in turn confers some computational advantages. A fully-annotated example illustrates the technique and presents some comparative numerical results.

Key Words: Optimization, Mathematical Programming, Reformulation, Relaxation, Variable Endogenization, Alkylation.

1 Introduction

It is well known that most optimization problems may be analysed in different ways—though typically as mathematical programs[2]—and that the various mathematical models may even share some numerical properties, but some may be easier to solve than others (for instance, in terms of the computational time needed to obtain the solution by the most efficient available solver). ‘Reformulations’—which the *Mathematical Programming Glossary* [4] defines as “obtaining a new formulation of a problem that is in some sense better, but equivalent to a given formulation”—are therefore quiet common in practice. Recasting a problem into the best possible formulation is a crucial aspect of any solution process and in this regard, the formal framework that has recently been proposed by Liberti [5] and applied by Costa [3] could prove useful.

The purpose of this paper is to highlight a reformulation that naturally “tumbles out” during the ‘variable endogenization’ process on equality-constrained M-programs, resulting in a partial relaxation of the constraints.[3] The exposition is by way of a fully annotated example; the numerical results were obtained by GENO—an evolutionary solver whose technical description beyond this footnote[4] is not prerequisite to the algorithm presented herein and is therefore excluded; otherwise, this paper is organised as follows: Part I of §**2** presents a mathematical model the optimization problem under consideration; Part II reformulates the model in Part I into a partially relaxed, box-constrained M-program via ‘variable endogenization’; some numerical results are presented and discussed in Part III; §**3** presents a summary and draws some conclusions; and finally, §**4** sets forth the legal framework governing this publication.

[1] The variable endogenization technique was briefly introduced in [15], and explained more substantively in [14].

[2] The abbreviation ‘MP’ or the contraction ‘M-program’ shall often be used in place of ‘mathematical program’ throughout this paper.

[3] See Step 6 of the algorithm in Part II of §**2**.

[4] GENO is an acronym for ***G***eneral ***E***volutionary ***N***umerical ***O***ptimizer. GENO is a real-coded evolutionary algorithm that can be used to solve uni- or multi-objective optimization problems; the problems presented may be static or dynamic in character; they may be unconstrained or constrained by functional equality or inequality constraints, coupled with set constraints on the variables; the variables themselves may assume real or discrete values in any combination. For a more detailed description and performance evaluation of GENO, see [13].

2 Illustrative Example

I. Preamble

Alkylation is a commonly used method for upgrading gasoline,[5] and a long-standing practical problem associated with the process concerns the determination of the optimal operating conditions. Sauer, Coville & Burwick [12] tackled this optimization problem via a series of linear mathematical programs; Bracken & McCormick [2] on the other hand formulated a nonlinear M-program; and Berna, Locke & Westerberg [1] later devised a simpler version of the Bracken-McCormick model. This paper addresses the alkylation optimization problem via the Berna-Locke-Westerberg (BLW) model. And in order to afford a passive comparison of GENO's performance with existing results by other solvers on the alkylation problem, the paper adopts the notation in [6] where the BLW model is stated thus:

MP1: $$\min_{\mathbf{x}} f(\mathbf{x}) = 5.04x_0 + x_1 + 3.36x_{12} + 0.35x_{13} - 6.3x_2x_4$$

Subject to:
$$1.22x_2 - x_0 - x_{12} = 0$$
$$0.98\,x_1 - x_3\,(x_1 + 0.01\,x_2x_6) = 0$$
$$x_{12} + 10x_{13} - x_0x_5 = 0$$
$$x_2x_8 - x_0(1.12 + 0.13167\,x_5 - 0.0067\,x_5^2) = 0$$
$$x_4x_9 - 0.57425 - 0.325\,x_3 + (0.0038\,x_5 - 0.01098)\,x_5 = 0$$
$$x_6x_{10} + 22.2\,x_7 - 35.82 = 0$$
$$1.33 + x_7x_{11} - 3x_4 = 0$$
$$x_0 \in [0, 2]$$
$$x_1 \in [0, 1.2]$$
$$x_2 \in [0, 5]$$
$$x_3 \in [0.85, 0.93]$$
$$x_4 \in [0.90, 0.95]$$
$$x_5 \in [3, 12]$$
$$x_6 \in [1.2, 4]$$
$$x_7 \in [1.45, 1.62]$$
$$x_8 \in [0.99, (1/0.99)]$$
$$x_9 \in [0.99, (1/0.99)]$$
$$x_{10} \in [0.9, (1/0.9)]$$
$$x_{11} \in [0.99, (1/0.99)]$$
$$x_{12} \in [0, 2]$$
$$x_{13} \in [0, 1.6]$$

[5] The alkylation process is explained in [7]; its goal is to improve the octane number of an olefin feed stream through a reaction involving isobutene and acid. The reaction product stream is distilled with a lighter hydrocarbon fraction recycled back to the reactor. An M-program formulation typically considers maximizing alkylate production less the material (i.e. feed stream) and operating (i.e. recycle) costs; the design parameters all take on continuous values and include the olefin feed rate (barrels/day), acid addition rate (thousands of pounds/day), alkylate yield (barrels/day), acid strength (wt. %), motor octane number, external isobutene to olefin ratio, and the performance number.

II. The Partial Relaxation Algorithm

Step 1: Re-write the original M-program in a script and using a text editor that allows character replacement, e.g. Microsoft Word^{TM}; label the constraints so as to avoid losing track in the manipulations to follow.

MP2: Min $f(\mathbf{X}) = 5.04*X0 + X1 + 3.36*X12 + 0.35*X13 - 6.3*X2*X4$

Subject to: $1.22*X2 - X0 - X12 = 0$ (i)

$0.98*X1 - X1*X3 - 0.01*X2*X3*X6 = 0$ (ii)

$X12 + 10*X13 - X0*X5 = 0$ (iii)

$X2*X8 - 1.12*X0 - 0.13167*X0*X5 + 0.0067*X0*X5*X5 = 0$ (iv)

$X4*X9 - 0.57425 - 0.325*X3 - 0.01098*X5 + 0.00038*X5*X5 = 0$ (v)

$X6*X10 + 22.2*X7 - 35.82 = 0$ (vi)

$1.33 + X7*X11 - 3*X4 = 0$ (vii)

$X0 \in [0, 2]$	$X1 \in [0, 1.2]$	$X2 \in [0, 5]$	$X3 \in [0.85, 0.93]$
$X4 \in [0.9, 0.95]$	$X5 \in [3, 12]$	$X6 \in [1.2, 4]$	$X7 \in [1.45, 1.62]$
$X8 \in [0.99, (1 / 0.99)]$	$X9 \in [0.99, (1 / 0.99)]$	$X10 \in [0.9, (1 / 0.9)]$	$X11 \in [0.99, (1 / 0.99)]$
$X12 \in [0, 2]$	$X13 \in [0, 1.6]$		

Step 2: Prepare a 'Connexion Matrix' (formally known as 'Incidence Matrix') that shows what variable appears in which constraint; shuffle the rows and columns such that a discernible ascending or descending "echelon of rows" emerges—this ensures non-circular definition of the endogenous Z-variables at the next step. Also, the variables constituting the slopping edge of the echelon (indicated by the red tick ✓ in Table 1) should be such that they can easily be made the subjects of the Z-equations at Step 5.

Table 1: The Connexion Matrix in 'Row Echelon' Form

	X8	X9	X1	X10	X13	X0	X4	X5	X2	X3	X6	X7	X12	X11	Notes
vii							✓					✓		✓	Def: Z1 ≡ X4
i						✓			✓				✓		Def: Z2 ≡ X0
iii					✓	✓		✓					✓		Def: Z3 ≡ X13
vi				✓							✓	✓			Def: Z4 ≡ X10
ii			✓						✓	✓	✓				Def: Z5 ≡ X1
v		✓					✓	✓		✓					Def: Z6 ≡ X9
iv	✓					✓		✓	✓						Def: Z7 ≡ X8
	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Y1	Y2	Y3	Y4	Y5	Y6	Y7	New Variables

Step 3: Re-define the X-variables that appear on the "edge" of the "echelon of rows" as sequentially numbered Z-variables and introduce sequentially numbered Y-variables for the remaining X-variables as shown in the bottom row of Table 1; in order to minimize coding errors, prepare a table of old and new variables as follows.

Table 2: Variable Replacement Table

Original Name	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13
Intermediate Name	Z2	Z5	-	-	Z1	-	-	-	Z7	Z6	Z4	-	-	Z3
New Name	Z2	Z5	Y2	Y3	Z1	Y1	Y4	Y5	Z7	Z6	Z4	Y7	Y6	Z3

Step 4: Use the ‘Replace All’ method to introduce the “new” Y and Z-variables into the *entire* M-program, viz.:

MP_3: Min $f(\mathbf{Y}, \mathbf{Z}) = 5.04*Z2 + Z5 + 3.36*Y6 + 0.35*Z3 - 6.3*Y2*Z1$

Subject to:

$1.22*Y2 - Z2 - Y6 = 0$ (i)

$0.98*Z5 - Z5*Y3 - 0.01*Y2*Y3*Y4 = 0$ (ii)

$Y6 + 10*Z3 - Z2*Y1 = 0$ (iii)

$Y2*Z7 - 1.12*Z2 - 0.13167*Z2*Y1 + 0.0067*Z2*Y1*Y1 = 0$ (iv)

$Z1*Z6 - 0.57425 - 0.325*Y3 - 0.01098*Y1 + 0.00038*Y1*Y1 = 0$ (v)

$Y4*Z4 + 22.2*Y5 - 35.82 = 0$ (vi)

$1.33 + Y5*Y7 - 3*Z1 = 0$ (vii)

$Y1 \in [3, 12]$ $Y2 \in [0, 5]$ $Y3 \in [0.85, 0.93]$ $Y4 \in [1.2, 4]$

$Y5 \in [1.45, 1.62]$ $Y6 \in [0, 2]$ $Y7 \in [0.99, (1 / 0.99)]$

$Z1 \in [0.9, 0.95]$ $Z2 \in [0, 2]$ $Z3 \in [0, 1.6]$ $Z4 \in [0.9, (1 / 0.9)]$

$Z5 \in [0, 1.2]$ $Z6 \in [0.99, (1 / 0.99)]$ $Z7 \in [0.99, (1 / 0.99)]$

Step 5: Solve each definitional Z-equation in MP_3 for the respective Z-variable and eliminate all other Z-variables appearing on the right-hand side by direct substitution using the appropriate Z-equation; re-write MP_3 as:

MP_4: Given:

$Z1 = (1.33 + Y5*Y7) / 3$ (vii)

$Z2 = 1.22*Y2 - Y6$ (i)

$Z3 = (1.22*Y2*Y1 - Y6*Y1 - Y6) / 10$ (iii)

$Z4 = (35.82 - 22.2*Y5) / Y4$ (vi)

$Z5 = 0.01*Y2*Y3*Y4 / (0.98 - Y3)$ (ii)

$Z6 = 3*(0.57425 + 0.325*Y3 + 0.01098*Y1 - 0.00038*Y1*Y1) / (1.33 + Y5*Y7)$ (v)

$Z7 = ((1.22*Y2 - Y6) / Y2)*(1.12 + 0.13167*Y1 - 0.0067*Y1*Y1)$ (iv)

Min $f(\mathbf{Y}, \mathbf{Z}) = 5.04*Z2 + Z5 + 3.36*Y6 + 0.35*Z3 - 6.3*Y2*Z1$

Subject to: $Y1 \in [3, 12]$ $Y2 \in [0, 5]$ $Y3 \in [0.85, 0.93]$ $Y4 \in [1.2, 4]$

$Y5 \in [1.45, 1.62]$ $Y6 \in [0, 2]$ $Y7 \in [0.99, (1 / 0.99)]$

$Z2 \in [0, 2]$ $Z5 \in [0, 1.2]$ $Z1 \in [0.9, 0.95]$ $Z7 \in [0.99, (1 / 0.99)]$

$Z6 \in [0.99, (1 / 0.99)]$ $Z4 \in [0.9, (1 / 0.9)]$ $Z3 \in [0, 1.6]$

Step 6: Note that Step 5 expresses the Z-equations solely in terms of the exogenous Y-variables, and when the said equations are substituted into the set bounds pertaining to the Z-variables, one obtains the following functional constraints:

$(1.33 + Y5*Y7) / 3 \in [0.9, 0.95]$ (vii)

$1.22*Y2 - Y6 \in [0, 2]$ (i)

$(1.22*Y2*Y1 - Y6*Y1 - Y6) / 10 \in [0, 1.6]$ (iii)

$(35.82 - 22.2*Y5) / Y4 \in [0.9, (1 / 0.9)]$ (vi)

$0.01*Y2* Y3*Y4 / (0.98 - Y3) \in [0, 1.2]$ (ii)

$3*(0.57425 + 0.325*Y3 + 0.01098*Y1 - 0.00038*Y1*Y1) / (1.33 + Y5*Y7) \in [0.99, (1 / 0.99)]$ (v)

$((1.22*Y2 - Y6) / Y2)*(1.12 + 0.13167*Y1 - 0.0067*Y1*Y1) \in [0.99, (1 / 0.99)]$ (iv)

And by substituting the Z-variables out of the objective function, the original program may be expressed as a smaller NLP featuring the *partially relaxed* functional constraints[6] listed above, viz.:

[6] As opposed to the “tight” equality constraints of the original program (see Remarks 1 below)

MP$_5$: Min f(**Y**) = 5.04*(1.22*Y2 – Y6) + (0.01*Y2*Y3*Y4 / (0.98 – Y3)) + 3.36*Y6 + \
0.35*((1.22*Y2*Y1 – Y6*Y1 – Y6) / 10) – 6.3*Y2*((1.33 + Y5*Y7) / 3)

Subject to: (1.33 + Y5*Y7) / 3 ∈ [0.9, 0.95] (i)

1.22*Y2 – Y6 ∈ [0, 2] (ii)

(1.22*Y2*Y1 – Y6*Y1 – Y6) / 10 ∈ [0, 1.6] (iii)

(35.82 – 22.2*Y5) / Y4 ∈ [0.9, (1 / 0.9)] (iv)

0.01*Y2* Y3*Y4 / (0.98 – Y3) ∈ [0, 1.2] (v)

3*(0.57425 + 0.325*Y3 + 0.01098*Y1 – 0.00038*Y1*Y1) / (1.33 + Y5*Y7) ∈ [0.99, (1 / 0.99)] (vi)

((1.22*Y2 – Y6) / Y2)*(1.12 + 0.13167*Y1 – 0.0067*Y1*Y1) ∈ [0.99, (1 / 0.99)] (vii)

Y1 ∈ [3, 12] Y2 ∈ [0, 5] Y3 ∈ [0.85, 0.93]

Y4 ∈ [1.2, 4] Y5 ∈ [1.45, 1.62] Y6 ∈ [0, 2]

Y7 ∈ [0.99, (1 / 0.99)]

Remarks 1. Generally speaking, a ‘relaxation’ of an M-program MP is an auxiliary problem of MP that has *fewer* constraints; relaxations are normally effected by simply ignoring certain constraints, variable bounds and / or variable types [5, p.15]. The fact that MP$_5$ has the same number of constraints as MP$_2$ may seem to disqualify it as a relaxation of the latter according to this definition. But if functional equality constraints are viewed as set-membership restrictions, i.e. $c(\mathbf{x}) = 0 \Leftrightarrow c(\mathbf{x}) \in \{0\}$, then ignoring a constraint is equivalent to expanding the constraining set on the RHS from the singleton {0} to the entire real line (-∞, ∞), i.e. a *total* relaxation of the constraining set. From that perspective, the algorithm presented above may be seen to be a ‘relaxation’ of sorts because it results in a formulation in which the original constraining set is expanded from {0} to [*a*, *b*]; its classification as a *partial relaxation* follows from the fact that the end-points of the new constraining set, i.e. *a* and *b*, are finite. And in terms of Liberti’s taxonomy [3, 5], the reformulation proffered here is an ‘opt-reformulation’ that preserves both local and global optimality information.

Note that the commonly used technique for dealing with functional equality constraints where $c(\mathbf{x}) = 0$ is replaced by two inequalities, $c(\mathbf{x}) \geq 0$ and $c(\mathbf{x}) \leq 0$, may seem to be a partial relaxation in the sense just described—but it is not. This is because the effective constraining set on $c(\mathbf{x})$ still remains the singleton {0}, and this is easily demonstrated as follows:

$$c(\mathbf{x}) \geq 0 \;\Leftrightarrow\; c(\mathbf{x}) \in [0, \infty) \qquad \text{(A)}$$

$$c(\mathbf{x}) \leq 0 \;\Leftrightarrow\; c(\mathbf{x}) \in (-\infty, 0] \qquad \text{(B)}$$

And the effective constraining set on $c(\mathbf{x})$ is a conjunction of the constraining sets in (A) and (B), i.e.:

$$[0, \infty) \cap (-\infty, 0] \equiv \{0\} \qquad \text{(C)}$$

The relaxation afforded by the endogenization technique would be complete by this stage, and normally one would proceed to solve MP$_5$ using any technique that takes advantage of such a formulation—which is certainly the case with most evolutionary algorithms. GENO’s solution for this formulation of the BLW model is depicted in Table 5a below.

But the nature of the new feasible region in MP$_5$ also allows use of the global optimization technique described in [14]. To see this, consider constraint (i), for example: because the constraining set on the RHS is a closed interval that lies entirely within the non-negative half line [0, ∞), one may introduce a positive slack variable and turn it into an equation as follows:

(1.33 + Y5*Y7) / 3 – W1 = 0, with W1 ∈ [0.9, 0.95]

The same applies to all the constraints in MP$_5$, and so provided one can evaluate the gradient of *f* with respect to the slack variable vector $\mathbf{W} \equiv (\text{W1, W2}, \ldots, \text{W7})^{\text{T}}$, then one may use the algorithm presented in [14] to compute the global solution, except that, in this case, recourse to interval analysis at Step 8 and / or Step 9 would be unnecessary because the bounds on the slack variables are already “tight”.

Alternatively, one may choose to perform further rounds of variable endogenization; however, this must be done in a systematic fashion as follows.

Step 7: Prepare a new ‘Connexion Matrix’ from MP_5 and shuffle the rows and columns into “row echelon” form.

Table 3: New Connexion Matrix

	Y5	Y2	Y7	Y6	Y1	Y4	Y3	Notes
i	✓		✓					Def: Z1 ≡ Y7
ii		✓		✓				Def: Z2 ≡ Y6
iii		✓		✓	✓			Def: Z3 ≡ Y1
iv	✓					✓		Def: Z4 ≡ Y4
vi	✓		✓		✓		✓	Def: Z5 ≡ Y3
v		✓				✓	✓	
vii		✓		✓	✓			
	X1	X2	Z1	Z2	Z3	Z4	Z5	New Variables

Step 8: Introduce a set of slack variables into the “edge constraints” to turn them into equations and include the set-constraints pertaining to the slack variables into the M-program, viz.:

MP_6: Min $f(\mathbf{Y}, \mathbf{W}) = 5.04*(1.22*Y2 - Y6) + (0.01*Y2*Y3*Y4 / (0.98 - Y3)) + 3.36*Y6 + \backslash$
$0.35*((1.22*Y2*Y1 - Y6*Y1 - Y6) / 10) - 6.3*Y2*((1.33 + Y5*Y7) / 3)$

Subject to: $(1.33 + Y5*Y7) / 3 - W1 = 0$ (i)

$1.22*Y2 - Y6 - W2 = 0$ (ii)

$(1.22*Y2*Y1 - Y6*Y1 - Y6) / 10 - W3 = 0$ (iii)

$(35.82 - 22.2*Y5) / Y4 - W4 = 0$ (iv)

$0.01*Y2* Y3*Y4 / (0.98 - Y3) \in [0, 1.2]$ (v)

$3*(0.57425 + 0.325*Y3 + 0.01098*Y1 - 0.00038*Y1*Y1) / (1.33 + Y5*Y7) - W5 = 0$ (vi)

$((1.22*Y2 - Y6) / Y2)*(1.12 + 0.13167*Y1 - 0.0067*Y1*Y1) \in [0.99, (1 / 0.99)]$ (vii)

$Y1 \in [3, 12]$

$Y2 \in [0, 5]$

$Y3 \in [0.85, 0.93]$

$Y4 \in [1.2, 4]$

$Y5 \in [1.45, 1.62]$

$Y6 \in [0, 2]$

$Y7 \in [0.99, (1 / 0.99)]$

$W1 \in [0.9, 0.95]$

$W2 \in [0, 2]$

$W3 \in [0, 1.6]$

$W4 \in [0.9, (1 / 0.9)]$

$W5 \in [0.99, (1/0.99)]$

Step 9: Solve each W-induced equation for the Z-variables and re-state MP_6 as:

MP_7: Given:

$$Z1 \equiv Y7 = (3*W1 - 1.33) / Y5 \quad \text{(i)}$$

$$Z2 \equiv Y6 = 1.22*Y2 - W2 \quad \text{(ii)}$$

$$Z3 \equiv Y1 = (10*W3 + Y6) / (1.22*Y2 - Y6) \quad \text{(iii)}$$

$$Z4 \equiv Y4 = (35.82 - 22.2*Y5) / W4 \quad \text{(iv)}$$

$$Z5 \equiv Y3 = ((1.33 + Y5*Y7)*W5 - 3*(0.57425 + 0.01098*Y1 - 0.00038*Y1*Y1)) / 0.975 \quad \text{(vi)}$$

$$\text{Min } f(\mathbf{Y}) = 5.04*(1.22*Y2 - Y6) + (0.01*Y2*Y3*Y4 / (0.98 - Y3)) + 3.36*Y6 + \backslash$$
$$0.35*((1.22*Y2*Y1 - Y6*Y1 - Y6) / 10) - 6.3*Y2*((1.33 + Y5*Y7) / 3)$$

Subject to:

$$0.01*Y2*Y3*Y4 / (0.98 - Y3) \in [0, 1.2] \quad \text{(v)}$$

$$((1.22*Y2 - Y6) / Y2)*(1.12 + 0.13167*Y1 - 0.0067*Y1*Y1) \in [0.99, (1 / 0.99)] \quad \text{(vii)}$$

$Z3 \in [3, 12]$; $Y2 \in [0, 5]$; $Z5 \in [0.85, 0.93]$; $Z4 \in [1.2, 4]$

$Y5 \in [1.45, 1.62]$; $Z2 \in [0, 2]$; $Z1 \in [0.99, (1 / 0.99)]$

$W1 \in [0.9, 0.95]$ $W2 \in [0, 2]$ $W3 \in [0, 1.6]$ $W4 \in [0.9, (1 / 0.9)]$

$W5 \in [0.99, (1/0.99)]$

Step 10: Prepare a new variable replacement table and re-write MP_7 as shown below.

Table 4: Variable Replacement Table

Original Name	Y1	Y2	Y3	Y4	Y5	Y6	Y7	W1	W2	W3	W4	W5
Intermediate Name	Z3	-	Z5	Z4	-	Z2	Z1	-	-	-	-	-
New Name	Z3	X1	Z5	Z4	X2	Z2	Z1	X3	X4	X5	X6	X7

MP_8: Given:

$$Z1 = (3*X3 - 1.33) / X2 \quad \text{(i)}$$

$$Z2 = 1.22*X1 - X4 \quad \text{(ii)}$$

$$Z3 = (10*X5 + Z2) / (1.22*X1 - Z2) \quad \text{(iii)}$$

$$Z4 = (35.82 - 22.2*X2) / X6 \quad \text{(iv)}$$

$$Z5 = ((1.33 + X2*Z1)*X7 - 3*(0.57425 + 0.01098*Z3 - 0.00038*Z3*Z3)) / 0.975 \quad \text{(vi)}$$

$$\text{Min } f(\mathbf{X}) = 5.04*(1.22*X1 - Z2) + (0.01*X1*Z5*Z4 / (0.98 - Z5)) + 3.36*Z2 + \backslash$$
$$0.35*((1.22*X1*Z3 - Z2*Z3 - Z2) / 10) - 6.3*X1*((1.33 + X2*Z1) / 3)$$

Subject to:

$$0.01*X1*Z5*Z4 / (0.98 - Z5) \in [0, 1.2] \quad \text{(v)}$$

$$((1.22*X1 - Z2) / X1)*(1.12 + 0.13167*Z3 - 0.0067*Z3*Z3) \in [0.99, (1 / 0.99)] \quad \text{(vii)}$$

$Z3 \in [3, 12]$; $X1 \in [0, 5]$; $Z5 \in [0.85, 0.93]$; $Z4 \in [1.2, 4]$

$X2 \in [1.45, 1.62]$; $Z2 \in [0, 2]$; $Z1 \in [0.99, (1 / 0.99)]$

$X3 \in [0.9, 0.95]$ $X4 \in [0, 2]$ $X5 \in [0, 1.6]$ $X6 \in [0.9, (1 / 0.9)]$

$X7 \in [0.99, (1/0.99)]$

Remarks 2. At this stage, one may choose to eliminate the Z-variables from MP_8 by direct substitution and repeat the process starting at Step 7, but now focussing row and column manipulations on those variables that have not been internalised already, i.e. X1 and X2; in principle, this process may be repeated until a "core M-program" emerges, i.e. one whose connexion matrix cannot be manipulated into row-echelon form. But for the purposes of this paper, MP_8 is sufficient—GENO's solution for this formulation is depicted in Table 5b.

III. Numerical Results[7]

Table 5a: Solution of MP_5 by GENO

Generation Number	Time[8] (sec)	Objective
0	0.00	0.42605558
40	15.96	-1.76478895
80	16.29	-1.76478994
120	16.27	-1.76479147
160	16.19	-1.76495184
200	16.01	-1.76499551
240	16.19	-1.76499551
280	16.43	-1.76499709
320	16.18	-1.76499965
360	16.36	-1.76499965
400	16.27	-1.76499965
440	16.32	-1.76499965
480	16.38	-1.76499965

Optimal State Vector: $\mathbf{x} = (10.47546485, 3.03582191, 0.90131949, 1.56163636, 1.53535354, 2.00000000, 0.99000000)^T$

Objective Function Value: $f(\mathbf{x}) = -1.76499965$

Average execution time per 40 generations: 15.57 seconds

Overall execution time on 500 generations: 202.65 seconds

Approximate time to first optimum (for 3rd decimal point accuracy): 15.96 seconds

Table 5b: Solution of MP_8 by GENO

Generation Number	Time[9] (sec)	Objective
0	0.0000	0.46216603
20	1.9060	-1.15058452
40	1.7340	-1.15143366
60	1.8440	-1.15143474
80	1.7190	-1.15143484
100	1.6870	-1.15143484
120	1.7190	-1.15143539
140	1.7030	-1.15143558
160	1.7350	-1.15143558
180	1.7810	-1.76486901
200	1.8440	-1.76499920
220	1.7500	-1.76499924
240	1.7500	-1.76499924
260	1.7650	-1.76499924
280	1.7820	-1.76499924
300	1.7650	-1.76499924

Optimal State Vector: $\mathbf{x}^* = (3.03585360, 1.53535349, 0.94999999, 1.70374139, 1.58517072, 1.11111111, 0.99000000)^T$

Optimal Endogenous Vector: $\mathbf{z}^* = (0.99000000, 2.00000000, 10.47794419, 1.56163725, 0.90129642)^T$

Objective Function Value: $f(\mathbf{x}) = -1.76499924$

Average execution time per 20 generations: 1.7656 seconds

Overall execution time on 300 generations: 26.4840 seconds

Approximate time to first optimum (for 3rd decimal point accuracy): 15.8904 seconds

[7] The 'time-to-first-optimum' [15] is quoted for third decimal place accuracy in order to compare GENO's results with those reported in [6].

[8] The execution times pertain to a GAUSS version of GENO running under Windows 7 on a Desk Top machine with the following specs: AMD Phenom^{TM} II x6 1055T Processor, 2.81GHz, 8GB RAM. The mating population was of size 60.

[9] The execution times pertain to a C++ version of GENO running under Windows 8.1 on a Laptop machine with the following hardware specs: AMD A4-5000 APU Processor, 1.5GHz, 4GB RAM. The mating population was of size 40.

Remarks 3. The BLW model is one approach at solving a practical alkylation problem whose global solution is unknown—its best approximation thus far remains that first reported by Berna, *et al*. [1], namely –1.765. The results depicted in Table 6 below are those reported in a recent evaluation of several NLP solvers on the alkylation problem conducted by Loehle [6], to which are appended GENO's results for comparison. The NLP solvers in Loehle's study employ a variety of numerical methods (see [6, 9] for brief descriptions); these were as follows:

1. `FindMinimum`, `NMinimize` and `Minimize` in *Mathematica 5.1*;[10]
2. `MS`, `CNLP` and `Optimize` in *MathOptimizer*;[11]
3. `GlobalSearch` and `GlobalPenaltyFn` in *Global Optimization*.[12]

The GENO results included in Table 6 are those from Table 5a only.

Table 6: Alkylation Optimization: GENO versus other NLP solvers

ALGORITHM	MINIMUM VALUE	TIME (sec)	REMARKS
GlobalSearch	-1.756	24.7	
GlobalPenaltyFn	-0.990	655	
NMinimize	-1.390	8.8	
NMinimize_SA	-1.359	6.8	
NMinimize_DE	-1.390	6.9	
NMinimize_RS	-1.320	168.9	
Minimize	-1.390	8.7	
CNLP	-	6.4	Infeasible solution
MS	-	63.9	Infeasible solution
Optimize	-	220	Infeasible solution
GENO	-1.765	15.96	Solved in 40 generations

Table 6 clearly shows that most of the NLP solvers in Loehle's study performed poorly on the alkylation problem,[13] and a possible explanation as to why this was so may largely be due to the nature of the feasible region. To see this, let the vector equation **C**(**x**) = **0** be the entire constraint system in an M-program under consideration, and let ζ denote the collection of points (in decision space) that solve the system **C**(**x**) = **0**. By definition, the feasible set is precisely the set ζ. Usually ζ is can be expected to be countable and finite for well-posed equation systems, but it is not uncommon for the size of ζ to be greater than one—i.e. the multiple solutions case—or even empty. Assuming ζ is not empty but discrete and finite—and irrespective of the numerical technique employed by the solver—the primary task of any solution algorithm is, in essence, to generate a sequence { $\mathbf{x}_k$ } that firstly converges onto the feasible region, i.e. one element of ζ in this case, and secondly optimizes the criterion function. Clearly, the sparser ζ is, the more difficult it is for the solution algorithm to target the (isolated) optimal point in ζ.

By contrast, the feasible region in a partially relaxed program such as MP5 is contiguous and extensive, it being the Cartesian product of the constraining sets [a_i, b_i] on the right-hand side of each functional constraint, viz.:

$$\tilde{\mathbf{C}}(\mathbf{y}) \in \prod_{i=1}^{m} [a_i, b_i] \qquad \text{(D)}$$

Because targeting the feasible set in (D) is much easier, the overall solution process of a partially relaxed program can be expected to be more efficient; the results by GENO on the relaxed formulations appear to support this notion.

[10] Authored by Wolfram Research Inc.

[11] Authored by Pinter Consulting Inc.

[12] Authored by Loehle Enterprises LLC.

[13] But Pintér & Kampas [8] report a vastly improved performance by a revised version of the *MathOptimizer Professional* package; in fact, the core solver suite has been further improved since publication of the 2013 Pintér & Kampas paper [11]; the solver's latest performance profile may be found in [10].

3 Summary and Conclusions

Reformulations are modelling activities aimed at obtaining an alternative mathematical specification of a problem that is in some sense better than a given formulation; they are very common in optimization practice. This paper has presented a reformulation for equality-constrained M-programs that is a natural “by-product” of the ‘variable endogenization’ process; the resulting “new” model involves *partially relaxed* constraints that confer computational advantages, at least for evolutionary algorithms—numerical results obtained by GENO on the alkylation optimization problem support this assertion. Assuming the connexion matrix permits it, the relaxation technique proffered here is simple to implement and worth trying out in practice.

4 Legalities

I. Licence and Trademarks

Except for the trademark items mentioned below, this work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit this page;[14] to override specific prohibitions of the governing licence,[15] submit a formal request to the author at: ike_siwale@hotmail.com.

GENO^{TM} is a trademark of Apex Research Ltd
Copyright © 1997-2015
All Rights Reserved Worldwide

GAUSS^{TM} is a trademark of Aptech Systems Inc.
Copyright © 1983-2015
All Rights Reserved Worldwide

II. Disclaimer

This document contains proprietary material created by Apex Research Ltd which is subject to further verification and change without notice; however, Apex Research Ltd is under no obligation to provide an updated version. Furthermore, Apex Research Ltd does not make any expressed or implied warranty— including the warranties of merchantability and fitness for a particular purpose—as to the accuracy or completeness of the methods described herein; accordingly, Apex Research Ltd accepts no liability for any damages that may occur from use.

[14] http://creativecommons.org/licenses/by-nc-sa/4.0/

[15] http://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

References

1. Berna, T., Locke, M., Westerberg, A. W.: A new approach to optimization of chemical processes. *American Institute of Chemical Engineers Journal*, **26**, 37-43 (1980).

2. Bracken, J., McCormick, G. P.: *Selected Applications of Nonlinear Programming*. John Wiley, New York (1968)

3. Costa, A.: Applications of reformulations in mathematical programming. Unpublished Ph.D. Dissertation, Ecole Polytechnique X, (2012).

4. Holder, A. (ed.).: Mathematical programming glossary. *INFORMS Computing Society* (2006). [Online] Available at: http://glossary.computing.society.informs.org/ver2/mpgwiki/index.php?title=Main_Page

5. Liberti, L.: Reformulations in mathematical programming: Definitions and systematics. RAIRO-OR, **43**, 55–86 (2009)

6. Loehle, C.: Global optimization using *Mathematica*: A test of software tools. *Mathematica in Education and Research*, **11**, 139-152 (2006).

7. Payne, R. E.: Alkylation—What you should know about this process. *Petroleum Refiner* **37**, 316-329 (1958).

8. Pintér, J. D., Kampas, F. J.: Benchmarking nonlinear optimization software in technical computing environments. *Top*, **21**, 133-162 (2013).

9. Pintér, J. D., Kampas, F. J.: Model development and optimization with *Mathematica*. In: Golden, B. L., Raghavan, S., Wasil, E. A. (eds.).: *The Next Wave in Computing, Optimization and Decision Technologies*, pp. 285-302. Springer, New York (2005)

10. Pintér, J. D.: How difficult is nonlinear optimization? A practical solver tuning approach with illustrative results. (2014). [Online] Available at: http://www.optimization-online.org

11. Pintér, J. D.: Private communication (2015).

12. Sauer, R. N., Coville, A. R., Burwick, C. W.: Computer points the way to more profits. *Hydrocarbon Processing and Petroleum Refiner*, **43**, 84-92 (1964).

13. Siwale, I.: GENO^{TM} 2.0: The GAUSS User's Manual. *Tech. Rep. RD-13-2013*, Apex Research Ltd, London (2013). Available with a trial version of GENO—contact Aptech Systems Inc.

14. Siwale, I.: Global optimization via slack variables. *Tech. Rep. RD-16-2013*, Apex Research Ltd, London (2014). [Online] Available at: http://www.researchgate.net or at: http://www.optimization-online.org

15. Siwale, I.: Solution of nonlinear equations via optimization. *Tech. Rep. RD-15-2013*, Apex Research Ltd, London (2013). [Online] Available at: http://www.researchgate.net or at: http://www.optimization-online.org