SOLUTION OF NONLINEAR EQUATIONS VIA OPTIMIZATION

Isaac Siwale

Technical Report RD-15-2013
Apex Research Limited
London
England

e-mail: ike_siwale@hotmail.com

Abstract

This paper presents four optimization models for solving nonlinear equation systems. The models accommodate both over-specified and under-specified systems. A variable endogenization technique that improves efficiency is introduced, and a basic comparative study shows that the optimization methods presented are very effective.

Key Words: Optimization, Multiple Objective Programming, Nonlinear Equations, Game Theory, Compromise Solution, Evolutionary Algorithms.

1 Introduction

The need to solve systems of nonlinear equations arises in a diverse set of applied science and engineering fields: for example, the test suite by Averick, Carter, More & Xue [**1**] is on practical problems emanating from fluid dynamics, medicine, elasticity, combustion, molecular conformation, chemical kinetics, lubrication and superconductivity. But in the book *Numerical Recipes in C*: *The Art of Scientific Computing*, the authors assert:

> "We make an extreme, but wholly defensible, statement: There are *no* good general methods for solving systems of more than one non-linear equation. Furthermore, it is not hard to see why (very likely) there *never will* be any good, general methods." [Paraphrased from **20**, p.379; emphasis in original]

And this is not an isolated view—Dennis & Schnabel also express similar sentiments in their book [**4**, p.16].

But such "pessimism" is only defensible if one restricts one's attention to traditional deterministic algorithms. When other solution methodologies are taken into account, then the situation is not so bleak: algorithms that combine the calculus of real intervals [**17**] and / or constraint satisfaction techniques with standard Newton-type and homotopy algorithms have since been developed and shown to be robust; stochastic algorithms that show a great deal of potential have also emerged; and last but not least, various formulations that alleviate the difficulties associated with the nonlinear equation system problem have been proposed. In support, one may cite Maranas & Floudas [**15**] who proposed a global optimization formulation solved by the branch-and-bound technique that can find all the solutions of nonlinear equation systems; Van Hentenryck, McAllester & Kapur [**30**] who present a branch-and-prune constraint satisfaction algorithm that behaves well on a variety of benchmarks; Basirzadeh, Kamyad & Effati [**2**] who propose a technique called the 'optimal time method' that employs measure theory in conjunction with optimal control theory; Grosan & Abraham [**9**] who present an evolutionary multi-objective optimization approach—a technique which this paper elaborates upon and hopefully clarifies; Nguyen Huu & Tran Van [**18**] who propose a probability-driven method; Ji, Wu, Li & Feng [**10**] who apply interval calculus techniques to a homotopy based algorithm in the framework of semi-algebraic systems; and Rahimian, Jalai, Seader & white [**21**] who propose a new homotopy method.

The purpose of this paper is to contribute other ideas for solving nonlinear equation systems. Occasionally, the exposition mentions a solver called GENO which was used to generate the numerical results reported, but whose description beyond this footnote[1] is not necessary for the arguments presented herein and is therefore excluded; otherwise, the paper is organised as follows: §2 reformulates an equation problem as a multi-objective programming problem; §3 delineates the character of the solution to the model in §2, and introduces the compromise solution concept; §4 re-formulates the model of §2 using distance metrics, and §5 presents a uni-objective version of the same; §6 presents a totally different approach based on NCP-functions and a saddle point theorem; numerical examples are in §7; §8 summarises and concludes the presentation; last but not least, the legal framework governing this publication is set forth in §9.

2 Equation Systems: An Optimization Formulation

Nocedal & Wright [**19**] classify algorithms for solving nonlinear equation systems into three categories: (i) Newton and quasi-Newton methods; (ii) merit function approaches; (iii) continuation / homotopy techniques. But the same authors also point out that none of these methods are totally satisfactory in practice:

> "Newton-based methods all suffer from one shortcoming: unless the Jacobian matrix is non-singular in the region of interest—a condition that often cannot be guaranteed—they are in danger of converging to a local minimum of the merit function [associated with the equation system] that is not a solution to the system; continuation methods may fail to produce a solution even to a fairly simple system of nonlinear equations; however, they are more robust than merit-function based methods but are also computationally more expensive" [Paraphrased from **19**, pp. 296-301]

Alternative solvers for nonlinear equation systems are therefore required, and to that end, it is shown below that the very essence of constrained optimization affords one such method; the exposition is intentionally formal with emphasis on the rationale underlying particular formulations—it employs the following notation:

> **Notation.** A distinction is made between the *criterion set* and the *criterion vector* of a multi-objective optimization problem—the former is simply a discrete collection of criterion functions; the latter is the vector whose components are the elements of the criterion set. 'Opt' denotes an operator whose operand is generally a discrete set of criterion functions, i.e. the *criterion set*; it is a command to 'optimize the operand'; it may be "distributed onto" the elements of the criterion set, or "factored out" from such a collection; when the operand is a singleton, then 'Opt' means 'minimize' or 'maximize' depending on the context. The associated term 'arg opt' means 'the argument that optimizes the operand'; when the operand is a singleton, it means 'arg min' or 'arg max' depending on the context.

Let $\mathbf{x} \in \mathbf{R}^n$ be a decision vector whose j-th component is in $[L_j, U_j]$; let $\mathbf{C}(\mathbf{x}) : \mathbf{R}^n \to \mathbf{R}^m$ be a mapping whose components are nonlinear functions $c_i(\mathbf{x}) : \mathbf{R}^n \to \mathbf{R}$, $i \in \{1, 2, \ldots, m\}$; let f_0 be a numeric function that maps $\mathbf{R}^n$ into $\mathbf{R}$, and consider a generic mathematical program defined on the set $\mathbf{X}_1 \equiv \{\mathbf{x} \mid \mathbf{C}(\mathbf{x}) \geq 0\}$, viz.:

$$\text{MP}_1\text{:} \qquad \operatorname*{Opt}_{\mathbf{x}} \left\{ f_0(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}_1 \subset \mathbf{R}^n \right\}$$

Let $\mathbf{P}$ denote the proposition 'The vector $\mathbf{x}^* \in \mathbf{X}_1$ is a solution to MP_1'; then formally, we have that:

$$\mathbf{P} \;\Leftrightarrow\; \{\, \{\mathbf{x}^* = \arg\operatorname{opt} f_0(\mathbf{x})\} \wedge \{c_1(\mathbf{x}^*) \geq 0\} \wedge \{c_2(\mathbf{x}^*) \geq 0\} \wedge \ldots \wedge \{c_m(\mathbf{x}^*) \geq 0\} \,\} \tag{1a}$$

[1] 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 [**25**].

The task of any solution algorithm is to generate a sequence $\{ \mathbf{x}_k \}$ that converges onto a candidate solution point $\mathbf{x}^*$ that is within the feasible region *and* optimizes the primary criterion f_0. Note that each 'feasibility term' in the conjunction on the right-hand side of (1a) may be re-stated as $\{c_i(\mathbf{x}^*) \in [0, \infty)\}$, and the search for a feasible $\mathbf{x}^*$ may be viewed as a process that seeks to minimize the distance of each outcome $c_i(\mathbf{x}_k)$ from the set $[0, \infty)$—the said distance may be measured by the following metric:

$$f_i(\mathbf{x}) \equiv \begin{cases} 0, & \text{if } \mathbf{x} \text{ is feasible} \\ |c_i(\mathbf{x})|, & \text{if } \mathbf{x} \text{ is infeasible} \end{cases}, \quad i \in \{1, 2, \cdots, m\} \tag{1b}$$

It follows immediately that:

$$\mathbf{P} \Leftrightarrow \{ \{\mathbf{x}^* = \arg\,\text{opt}\, f_0(\mathbf{x})\} \wedge \{\mathbf{x}^* = \arg\,\min f_1(\mathbf{x})\} \wedge \ldots \wedge \{\mathbf{x}^* = \arg\,\min f_m(\mathbf{x})\} \} \tag{1c}$$

Thus the program MP1 is equivalent to an unconstrained multi-objective program, viz.:

$$\underset{\mathbf{x}}{\text{Opt}} \{ f_0(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}_1 \subset \mathbf{R}^n \} \Leftrightarrow \underset{\mathbf{x}}{\text{Opt}} \{ f_0(\mathbf{x}), f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}) \} \tag{1d}$$

Applying the 'constraint conversion' idea to the solution of equation systems is straight forward: it entails viewing equations as set-membership, i.e. $\mathbf{C}(\mathbf{x}) \in \{\mathbf{0}\}$, and notionally embedding the same into a program of the form MP1 whose primary criterion function f_0 is a constant (which may therefore be omitted from the optimization process). Note that the feasible set $\mathbf{X}_1$ includes the singleton $\{\mathbf{0}\}$, and so the embedding is fully justified. But although the method converts functional constraints into a *criterion set*, it is advantageous to still retain the former in the multi-objective formulation because this has the added effect of reinforcing the search for the solution—this is the approach adopted henceforth.

To illustrate, let $\mathbf{C}(\mathbf{x}) : \mathbf{R}^n \to \mathbf{R}^m$ be a vector-valued mapping whose components are functions denoted by $c_i(\mathbf{x}) : \mathbf{R}^n \to \mathbf{R}$, $i \in \{1, 2, \ldots, m\}$, at least one of which is nonlinear, and consider the generic uni-objective constrained mathematical program defined on the feasible set $\mathbf{X}_2 \equiv \{ \mathbf{x} \mid \mathbf{C}(\mathbf{x}) \in \{\mathbf{0}\} \}$:

MP2: $\quad \underset{\mathbf{x}}{\text{Opt}} \{ f_0(\mathbf{x}) = 0 \mid \mathbf{x} \in \mathbf{X}_2 \subset \mathbf{R}^n \}$

By the argument presented above, the program MP2 is equivalent to a multi-objective program, viz.:

$$\underset{\mathbf{x}}{\text{Opt}} \{ f_0(\mathbf{x}) = 0 \mid \mathbf{x} \in \mathbf{X}_2 \subset \mathbf{R}^n \} \Leftrightarrow \underset{\mathbf{x}}{\text{Opt}} \{ f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), \cdots, f_m(\mathbf{x}) \mid \mathbf{C}(\mathbf{x}) \in \{\mathbf{0}\} \} \tag{1e}$$

Since we are only concerned with infeasible decision vectors (because all the feasible ones are, by definition, already solutions to $\mathbf{C}(\mathbf{x}) = \mathbf{0}$), we only need to consider the bottom branch of the distance functions in (1b), and therefore one may re-state the criterion set on the right-hand side of (1e) as $\{|c_1|, |c_2|, \ldots, |c_m|\}$. Thus the nonlinear vector equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ may be solved using the following multi-objective optimization problem in which the operator 'Opt', when "distributed" over the criterion set, denotes the command 'minimize':

MPem1: $\quad \underset{\mathbf{x}}{\text{Opt}} \{ |c_1(\mathbf{x})|, |c_2(\mathbf{x})|, \cdots, |c_m(\mathbf{x})| \}$

Subject to: $\quad c_i(\mathbf{x}) = 0; \;\; x_j \in [L_j, U_j], \;\; \forall i \in \{1, 2, \ldots, m\}; \;\; \forall j \in \{1, 2, \ldots, n\}$

Remarks 1. The formulation allows the number of variables, i.e. the dimension of the decision vector **x**, to be different from the total number of equations, all of which may be nonlinear; thus, just as the method in [**15**], both over-specified and under-specified linear and nonlinear systems are accommodated. Although the method converts the equations $c_i(\mathbf{x})$ into objectives, MP_{em1} still retains these as constraints for computational expediency under GENO. This technique is certainly not new as suggested by [**9**]—earlier proponents include Surry, Radcliffe & Boyd [**28**], Coello Coello [**3**], Siwale [**23**] and Klamroth & Jørgen [**11**]; in fact, one could argue that it is merely a "reverse" of well known multi-objective scalarization methods; see review in [**16**].

3 Notes on MP_{em1}

3.1 Preamble

Multi-objective problems such as MP_{em1} may be analysed in two related spaces: (i) the *decision space* is the set of all possible values of the decision vector **x**; components of the *n*-vector **x** are typically constrained to closed intervals on the real line, i.e. $x_i \in [L_i, U_i]$, and the decision space is the Cartesian product of such intervals; it includes the *feasible set*—i.e., $\mathbf{X}_2$ in the case of MP_{em1}—as a subset; (ii) the *outcome* or *criterion space* is the set of all criterion vectors that correspond to the decision vectors; the structure of the outcome space is determined by the nature of the criterion functions; a subset that is of particular interest is the *outcome set* which we define as the collection of all vectors in criterion space that correspond to *feasible* vectors in the decision space. Whereas uni-objective optimization is typically studied in the decision space, multi-objective programming is mostly studied in outcome space. Different types of solution may be defined and the various notions in this regard are discussed fully elsewhere [**26**]; this paper is only concerned with one, namely the compromise solution—its definition and computation are presented in §3.4 below.

3.2 Characterizing the Solution in Outcome Space

In outcome space solutions are ordered by a dominance relation attributed to Vilfredo Pareto (1848-1923), hence the descriptor 'Pareto-optimality'. Briefly stated, a vector **a** is said to Pareto-dominate another vector **b** in the 'greater-than' sense if the difference vector **d** = **a** – **b** only has non-negative elements, at least one of which is strictly positive; the applicable difference vector for dominance in the 'less-than' sense is **d** = **b** – **a**. The set of all non-dominated outcome vectors constitutes the 'Pareto frontier', and a point from this set may be selected (by some criteria) as the final solution to the multi-objective problem. In the case of MP_{em1} the Pareto-frontier may be ascertained as follows: assume each element $|c_i|$ of the criterion set $\{|c_1|, |c_2|, \ldots, |c_m|\}$ is finite for all decision vectors **x** and let S_i denote its supremum; then the outcome set is given by:

$$\mathbf{B} = \left\{ \mathbf{C} = (c_1, c_2, \cdots, c_m)' \in \mathbf{R}^m \;\middle|\; |c_i| \in [0, S_i) \right\} \tag{2}$$

If $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ has a solution, then there exists at least one $\mathbf{x}^*$ which is such that $\forall i$, $|c_i(\mathbf{x}^*)| = 0$;[2] in other words, the outcome set $\mathbf{B}$ must have a vertex at the origin of $\mathbf{R}^m$. The Pareto frontier in this case is a singleton, namely the point **0**, this being *the one and only point* that is not dominated by any other in $\mathbf{B}$. The system $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ has no solution if $\exists i \in \{1, 2, \ldots, m\}$: $\forall \mathbf{x}$, $c_i(\mathbf{x}) \in (0, S_i)$.

[2] Assuming a prior definition of 'zero' in terms of the location of the most significant digit

3.3 Characterizing the Solution in Decision Space

Decision vectors that map onto the Pareto frontier are said to be 'efficient'. In the case of MP_{em1}, one may describe its efficient set in decision space as follows: let ζ_i denote the set of roots (or zeros) of the *i*-th equation $c_i(\mathbf{x}) = 0$, i.e. $\zeta_i = \{\mathbf{x} : c_i(\mathbf{x}) = 0\}$; and let ζ denote the collective set of zeros of the system $\mathbf{C}(\mathbf{x}) = \mathbf{0}$. Usually, each ζ_i is a discrete set which may or may not be finite, and the solution set ζ is given by:

$$\zeta = \bigcap_{i=1}^{m} \zeta_i \tag{3}$$

The equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ will have no solution if at least one of the ζ_i sets is empty. But assuming a solution exists, there is still no easy way knowing the cardinality of the solution set ζ; and unlike the corresponding set in outcome space, it is not uncommon for the size of ζ to be greater than one, i.e. the multiple solutions case. However, for well posed equation systems, the set ζ is usually countable and finite, and the goal of all solution algorithms for $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ is to generate a sequence $\{\mathbf{x}_k\}$ that converges onto *at least one* element of ζ.

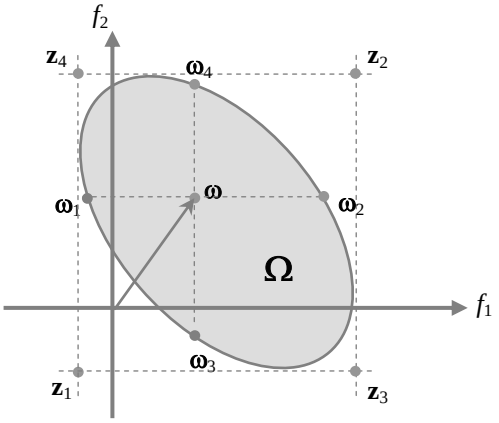
3.4 The Compromise Solution and its Computation

The peculiar nature of the Pareto-set, i.e. a singleton, suggests that algorithms that rely solely on the Pareto-dominance concept in computing the solution to MP_{em1} may be found wanting in this case because it is much more difficult to converge to a specific single point as opposed to an extensive Pareto frontier.[3] The Pareto-dominance criterion is certainly necessary but it may not be sufficient to ensure efficient convergence towards the Pareto set in this case. What is required—in addition to the dominance test—is a mechanism that, in effect, actively "pulls" candidate solutions towards the ideal outcome. The compromise solution concept embodies such a dual role and its proper implementation should achieve the desired end.

The compromise solution concept that was first introduced by Salukvadze [**22**] and later independently presented by Yu [**31**] and Zeleny [**32**]. It is based on the common-sense notion that the best option is a feasible point that yields criterion values that are closest to an ideal outcome—the *ideal* being that point at which each criterion is optimized to the fullest extent possible, i.e. the *global* solution. The rationale for the compromise solution is best explained in terms of the two-dimensional outcome space depicted in Figure 1 below in which $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are finite-valued criterion functions of a decision vector $\mathbf{x}$, and the collection of all such feasible outcomes constitutes the outcome set $\boldsymbol{\Omega}$. Associated with each outcome vector $\boldsymbol{\omega}$ are four 'boundary' outcome vectors $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, $\boldsymbol{\omega}_3$ and $\boldsymbol{\omega}_4$; these are points where lines that are parallel to the axes f_1 and f_2 intersect the boundary of the set $\boldsymbol{\Omega}$, which shall hereafter be denoted by $\partial\boldsymbol{\Omega}$. The vertices of the smallest rectangle enclosing $\boldsymbol{\Omega}$ comprise the 'utopia set', and for any given vertex vector $\mathbf{z}_n$, each of its dimensions represents the best possible outcome, i.e. the *global* solution, that could be attained by maximizing or minimizing a particular criterion independently. However, only one vertex would be relevant in any given scenario and such a vertex is called the 'ideal point'. In Figure 1 below, $\mathbf{z}_1$ is the ideal point when both criteria are required to be minimized; whereas $\mathbf{z}_4$ is for the case where criterion 1 is to be minimized, and criterion 2 maximized.

[3] Apart from this paper, Nguyen Huu & Tran Van [**18**, p.13] also confirm this in relation to the solutions reported by Grosan & Abraham [**9**]; they rightly point out that, although the Grosan-Abraham method is Pareto-dominance based, their solutions are *not* Pareto-optimal.

Figure 1: Outcome Set of a bi-objective Optimization Problem



One may define the compromise solution in two stages as follows:

- DEFINITION 1 [The Ideal Point]: Let $\boldsymbol{\Omega} \subset \mathbf{R}^n$ denote an outcome set; let L_j and U_j denote the lower and upper bounds respectively for the criterion f_j at $\boldsymbol{\omega}$ assuming all other outcomes remain constant; let $\mathbf{z}_i$, denote the *ideal point* for the problem at hand, then the coordinates of $\mathbf{z}_i$ are given by scalars z_{ij} defined as:

$$z_{ij} = \begin{cases} \underset{\omega \in \Omega}{\text{Sup}} \{U_j(\boldsymbol{\omega})\}, & \text{if the } j\text{th criterion requires maximizing} \\ \underset{\omega \in \Omega}{\text{Inf}} \{L_j(\boldsymbol{\omega})\}, & \text{if the } j\text{th criterion requires minimizing} \end{cases} \tag{4a}$$

  REMARKS 2: Because ideal outcomes are normally not *jointly* attainable, a compromise is required.

- DEFINITION 2 [The Tchebycheff Compromise Solution]: Let the point $\mathbf{z}_i$ be the *ideal point* a given multi-objective optimization problem; then the compromise solution is a member of those feasible controls whose outcomes are closest to the ideal outcome as measured by some distance metric in outcome space; thus, in terms of the Tchebycheff metric (see equation 5c below), the compromise solution is a feasible decision vector $\mathbf{x}^*$ whose corresponding outcome vector $\boldsymbol{\omega}^*$ belongs to a set of outcomes $\xi \subset \partial\boldsymbol{\Omega}$ that is defined as follows:

$$\xi(\mathbf{z}_i) = \{\boldsymbol{\omega} \in \partial\boldsymbol{\Omega} : \boldsymbol{\omega} = \arg\min \| \boldsymbol{\omega} - \mathbf{z}_i \|_\infty\} \tag{4b}$$

  REMARKS 3: The definition of $\xi(\mathbf{z}_i)$ entails two processes: (i) the obvious minimization process denoted by the 'arg min' operator; (ii) the less obvious search process that is supposed to delineate the boundary set $\partial\boldsymbol{\Omega}$. In the GENO scheme, the latter is approximated by evolutionary mechanisms using the Pareto-dominance criterion, and the former is a straight forward implementation of the Tchebycheff metric. The rationale underlying this solution concept may be explained as follows: (a) there is no question that, if it were achievable, the ideal outcome vector $\mathbf{z}_i$ would constitute *the* optimal solution to the multi-objective optimization problem under study; (b) but since this is usually not the case, one has to "compromise downwards" from the ideal outcome $\mathbf{z}_i$ to a less-than-ideal outcome $\boldsymbol{\omega}^*$ that corresponds to a feasible vector $\mathbf{x}^*$ and obviously, the extent of the "downward compromise", i.e. the quantity of criterion value that must be given up along each dimension, has to be minimal, hence the 'distance-minimizing' operation in the definition of the solution set ξ, and the stipulation that $\boldsymbol{\omega}^* \in \xi$

3.5 Closing Remarks

Equation systems may be recast into multi-objective optimization problems. For such a recast, the ideal outcome, the outcome set and the Pareto frontier all coalesce into one point—the origin of the outcome space. The compromise solution concept is most appropriate in this case.

4 The Set-of-Metrics Method

The basic multi-objective optimization model MP_{em1} is sufficient in many cases, but of course when the number of equations to be solved is large, then the speed of execution is inevitably degraded and convergence to the solution is unlikely to occur within a reasonable time frame. In order to achieve faster and higher quality performance, an alternative approach is required. To that end, it is well to remember that the criterion vector $\mathbf{J} \equiv (|c_1|, |c_2|, \ldots, |c_m|)^T$ resides in $\mathbf{R}^m$ — a real space in which the following metrics are well known:

$$d_1(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} |\, x_i - y_i \,| \tag{5a}$$

$$d_2(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5b}$$

$$d_\infty(\mathbf{x},\mathbf{y}) = \operatorname*{Max}_{i \in I} \{ |\, x_i - y_i \,| \} \tag{5c}$$

The functions in (5) are known as 'Manhattan', 'Euclidean' and 'Tchebycheff' metrics respectively, and it can be shown that if a sequence is convergent under any one of d_1, d_2, or d_∞, then it is convergent under the other two metrics as well, and the limits under the three metrics are the same [**29**, p.27]. The solution of the system $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ in outcome space is at the origin of $\mathbf{R}^m$, and the goal of all algorithms is to generate a sequence of criterion vectors $\{\mathbf{J}_k\}$ that converges onto $\mathbf{0}$. An obvious approach is to explicitly minimize some measure of the proximity of the vector $\mathbf{J}_k$ to $\mathbf{0}$ using the functions in (5), and one may include more than one metric in the proposed "proximity" objective function in order to exploit their different properties. Thus, instead of optimizing the set $\{|c_1|, |c_2|, \ldots, |c_m|\}$, one solves—in the most general case—the problem:

MP_{em2}: $\operatorname*{Opt}_{\mathbf{x}} \{ d_1(\mathbf{J},\mathbf{0}),\ d_2(\mathbf{J},\mathbf{0}),\ d_\infty(\mathbf{J},\mathbf{0}) \}$

Subject to: $c_i(\mathbf{x}) = 0;\ x_j \in [L_j, U_j],\ \forall i \in \{1, 2, \ldots, m\};\ \forall j \in \{1, 2, \ldots, n\}$

Remarks 4. The model MP_{em2} is only worth trying if the dimension m is significantly greater than 3, because there would be no saving in execution time otherwise. For that reason, no results by the model are presented and neither is model discussed any further in this paper, except to mention that it would be solved in exactly the same manner as MP_{em1}, i.e. via the compromise solution concept with the origin of $\mathbf{R}^3$ as the ideal point.

5 The Composite Metric Method

Experiments with GENO suggest that good results may be obtained from a single-objective version of MP_{em2} in which one optimizes a composite metric (denoted by d_c) made up of a *linear* combination of two or more of d_1, d_2, and d_∞. This paper proposes a simple sum—thus, in the most general case, one solves:

MP_{ed}: $\operatorname*{Min}_{\mathbf{x}} \{ d_c(\mathbf{J},\mathbf{0}) \equiv d_1(\mathbf{J},\mathbf{0}) + d_2(\mathbf{J},\mathbf{0}) + d_\infty(\mathbf{J},\mathbf{0}) \}$

Subject to: $c_i(\mathbf{x}) = 0;\ x_j \in [L_j, U_j],\ \forall i \in \{1, 2, \ldots, m\};\ \forall j \in \{1, 2, \ldots, n\}$

Remarks 5. Numerical tests show that the program MP$_{ed}$ even with only one metric included — the preferred option being d_∞ — is quite effective, at least on most problems in well known test suites for nonlinear equation systems that one may be find in [**30**], [**15**] and [**13**]. The model MP$_{ed}$ is closely related to those suggested by others: Maranas & Floudas [**15**] derive their global optimization model starting with a version of MP$_{ed}$ with only d_∞ present; the probability-based method of Nguyen Huu & Tran Van [**18**] effectively involves solving MP$_{ed}$ but without the constraints $c_i(\mathbf{x}) = 0$ and with only d_∞ retained.

6 The NCP Method

Alternatively, one could "embed" the equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ into a nonlinear complementarity problem and then solve the latter by optimization; the exposition requires some preliminary definitions as in [**7**]:

DEFINITION 3. Given a set $K \subset \mathbf{R}^n$ and a mapping $\mathbf{F} : K \to \mathbf{R}^n$, the variational inequality problem—hereafter denoted by VI(K, $\mathbf{F}$)—is to find the n-vector $\mathbf{x}^*$ in K such that:

VI(K, $\mathbf{F}$): $$\langle \mathbf{F}(\mathbf{x}^*), (\mathbf{y} - \mathbf{x}^*) \rangle \geq 0, \quad \forall \mathbf{y} \in K \tag{6a}$$

When the set K is restricted to the non-negative orthant of $\mathbf{R}^n$, then VI(K, $\mathbf{F}$) is equivalent to the nonlinear complementarity problem—denoted by NCP($\mathbf{F}$)—which may be stated thus: find $\mathbf{x}^*$ in K such that:

NCP($\mathbf{F}$): $$\mathbf{F}(\mathbf{x}^*) \geq \mathbf{0}; \qquad \mathbf{x}^* \geq \mathbf{0}; \qquad \langle \mathbf{F}(\mathbf{x}^*), \mathbf{x}^* \rangle = 0 \tag{6b}$$

Three index sets are of interest: $\alpha \equiv \{ i : x_i > 0 \}$; $\beta \equiv \{ i : f_i(\mathbf{x}) > 0 \}$; $\gamma \equiv \{ i : x_i = f_i(\mathbf{x}) = 0 \}$; the solution to NCP($\mathbf{F}$) is said to be non-degenerate if the third index set is empty.

NCP's are normally solved "indirectly", and a common approach is to introduce a bivariate function—referred to as 'the NCP-function'—that has a specific value when the relations that define an NCP are met. Formally, an NCP-function is a mapping $\phi : \mathbf{R}^2 \to \mathbf{R}$ for which the following statement holds:

$$\phi(a,b) = 0 \quad \Leftrightarrow \quad a \geq 0, \quad b \geq 0, \quad ab = 0 \tag{7}$$

NCP-functions are many and varied—a recent review these functions together with their properties may be found in [27]. The most common NCP-function is one named after Andreas Fischer and W. Burmeister but was actually constructed by the latter [**6**, p.271]; the Fischer-Burmeister NCP-function is defined on the non-negative orthant of $\mathbf{R}^2$ as follows:

$$\phi(a,b) = \sqrt{a^2 + b^2} - (a + b), \text{ with } a \geq 0 \text{ and } b \geq 0 \tag{8}$$

The Fischer-Burmeister function is positive homogeneous, Lipschitz-continuous and non-positive on $\mathbf{R}^2_+$; it attains its maximum value of zero when at least one of its arguments is zero, which also means that $ab = 0$.

In their approach to solving NCP's, Facchinei & Soares [**5**] introduce an auxiliary mapping that exploits the defining property of NCP-functions as follows: let 'vec$\{e_i\}$' denote a vector whose i-th element is e_i, and let ϕ denote an NCP-function; define the mapping $\mathbf{\Phi} : \mathbf{R}^n \to \mathbf{R}^n$ as follows:

$$\mathbf{\Phi}(\mathbf{x}) = \text{vec}\{ \phi(x_i, F_i(\mathbf{x})) \} \tag{9}$$

Consider the system of equations $\mathbf{\Phi}(\mathbf{x}) = \mathbf{0}$: by definition, each component of the vector in equation (9) is exactly the left-hand part of the 'NCP proposition' in (7); and since the right-hand part of the said proposition is exactly the same the scalar version of NCP(**F**), it follows that if $\mathbf{x}^*$ is a solution to $\mathbf{\Phi}(\mathbf{x}) = \mathbf{0}$, it is also a solution to NCP(**F**); and conversely, if $\mathbf{x}^*$ is a solution to NCP(**F**), then it is also a solution to $\mathbf{\Phi}(\mathbf{x}) = \mathbf{0}$. This is the basis of the solution method that is advocated below—the formulation is as follows. Let $\mathbf{C}(\mathbf{x}) : \mathbf{R}^n \rightarrow \mathbf{R}^m$ be a vector-valued mapping whose components are nonlinear functions $c_i(\mathbf{x}) : \mathbf{R}^n \rightarrow \mathbf{R}$, $i \in \{1, 2, \ldots, m\}$; let $\boldsymbol{\lambda} \in \mathbf{R}^m$ be a positive vector and let ϕ denote an NCP-function; define the function $\mathbf{\Phi} : \mathbf{R}^m \rightarrow \mathbf{R}^m$ as follows:

$$\mathbf{\Phi}(\boldsymbol{\lambda}, \mathbf{x}) = \text{vec}\{\ \phi(\lambda_i, c_i(\mathbf{x}))\ \} \tag{10a}$$

And finally, "embed" the equation system $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ into a pseudo-NCP as follows:[4]

$$\text{NCP}(\mathbf{C}): \qquad \boldsymbol{\lambda} \geq \mathbf{0}; \qquad \mathbf{C}(\mathbf{x}) \geq \mathbf{0}; \qquad \langle\ \boldsymbol{\lambda}, \mathbf{C}(\mathbf{x})\ \rangle = 0 \tag{10b}$$

Although equation (10b) is slightly different from the NCP definition given in (6b), it nonetheless also implies the following proposition:

$$\mathbf{\Phi}(\boldsymbol{\lambda}, \mathbf{x}) = \mathbf{0} \quad \Leftrightarrow \quad \{(\ \boldsymbol{\lambda} \geq \mathbf{0}\ ) \wedge (\ \mathbf{C}(\mathbf{x}) \geq \mathbf{0}\ ) \wedge (\ \langle\ \boldsymbol{\lambda}, \mathbf{C}(\mathbf{x})\ \rangle = 0\ )\} \tag{10c}$$

Thus, for *any non-negative* and *non-zero* vector $\boldsymbol{\lambda}^*$, the point $\mathbf{x}^*$ is a non-degenerate solution of the equation system $\mathbf{\Phi}(\boldsymbol{\lambda}^*, \mathbf{x}) = \mathbf{0}$ if and only if $\mathbf{x}^*$ is also a solution to NCP(**C**). One may compute the solution vector $\mathbf{x}^*$ by solving the equation system on the left-hand, or by solving the conjunctive relation on the right-hand side of (10c), perhaps as a constraint satisfaction problem; a common third alternative is to consider a suitably defined merit function $h : \mathbf{R}^n \rightarrow \mathbf{R}$ which provides a measure of the degree of coincidence between the solution of NCP(**C**) and that of an auxiliary program 'Min $h(\mathbf{x})$', and apparently " . . . it is not difficult to find a merit function for an NCP problem, the challenging task is to find a merit function that exhibits properties that are useful from a computational point of view" [**5**, p. 225].

However, the NCP method advocated in this paper is different; it is inspired by the fact that (10b) constitutes part of the well known Karush-Kuhn-Tucker optimality conditions for a nonlinear mathematical program of the form 'Min $\{f(\mathbf{x}) \mid \mathbf{C}(\mathbf{x}) \geq \mathbf{0}\}$'; it essentially combines the second and third methods mentioned above by minimizing the Lagrange function:

$$L_e\ (\boldsymbol{\lambda}, \mathbf{x}) \equiv f\ (\mathbf{x})\ -\ \langle\ \boldsymbol{\lambda}, \mathbf{C}(\mathbf{x})\ \rangle \tag{11a}$$

Following Kuhn & Tucker [**12**], one may characterize the solution to the vector equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ in terms of the Lagrangian $L_e\ (\boldsymbol{\lambda}, \mathbf{x})$ by notionally embedding it into a program of the form Min $\{f(\mathbf{x}) \mid \mathbf{C}(\mathbf{x}) \geq \mathbf{0}\}$, and then posing the following saddle-value problem.

- LAGRANGIAN SADDLE-VALUE PROBLEM: Let $\mathbf{x}$ be a decision vector in $\mathbf{R}^n$; $\boldsymbol{\lambda}$ be a non-negative vector in $\mathbf{R}^m$; and $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ be a vector equation of interest. Assuming the auxiliary program Min $\{f(\mathbf{x}) \mid \mathbf{C}(\mathbf{x}) \geq \mathbf{0}\}$ and the existence of a saddle point, find the pair $(\boldsymbol{\lambda}^*, \mathbf{x}^*)$ that results in a saddle value of the Lagrangian L_e, viz:

$$L_e\ (\boldsymbol{\lambda}, \mathbf{x}^*) \leq L_e\ (\boldsymbol{\lambda}^*, \mathbf{x}^*) \leq L_e\ (\boldsymbol{\lambda}^*, \mathbf{x}) \tag{11b}$$

[4] The prefix 'pseudo' is used because in a "true NCP", the vector $\boldsymbol{\lambda}$ would also be the sole argument of the function $\mathbf{C}(\bullet)$.

The following sufficiency theorem connects the saddle-value problem to the global minimizer of $f(\mathbf{x})$—a function yet to be defined—and the solution of $\mathbf{C}(\mathbf{x}) = \mathbf{0}$:[5]

□ THEOREM: Consider the auxiliary program: Min $\{f(\mathbf{x}) \mid \mathbf{C}(\mathbf{x}) \geq \mathbf{0}\}$. For any fixed, non-negative and non-zero vector $\boldsymbol{\lambda}^*$, if there exists a vector $\mathbf{x}^*$ such that the pair $(\boldsymbol{\lambda}^*, \mathbf{x}^*)$ solves the Lagrangian saddle value problem in (11b), then $\mathbf{x}^*$ is a *global* minimizer of $f(\mathbf{x})$ *and* a solution to the vector equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$.

□ PROOF: The left-hand inequality of the saddle value problem simplifies as follows:

$$f(\mathbf{x}^*) - \langle\, \mathbf{C}(\mathbf{x}^*), \boldsymbol{\lambda}\, \rangle \;\leq\; f(\mathbf{x}^*) - \langle\, \mathbf{C}(\mathbf{x}^*), \boldsymbol{\lambda}^*\, \rangle \tag{12a}$$

$$-\langle\, \mathbf{C}(\mathbf{x}^*), \boldsymbol{\lambda}\, \rangle \;\leq\; -\langle\, \mathbf{C}(\mathbf{x}^*), \boldsymbol{\lambda}^* \rangle \tag{12b}$$

$$\langle\, \mathbf{C}(\mathbf{x}^*), \boldsymbol{\lambda}\, \rangle \;\geq\; \langle\, \mathbf{C}(\mathbf{x}^*), \boldsymbol{\lambda}^*\, \rangle \tag{12c}$$

For an arbitrary fixed vector $\boldsymbol{\lambda}^*$, the inequality in (12c) is true for all $\boldsymbol{\lambda}$ if and only if $\mathbf{C}(\mathbf{x}^*) = \mathbf{0}$, i.e. when $\mathbf{x}^*$ is a solution to the equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$.[6]

Granted the fact that $\mathbf{C}(\mathbf{x}^*) = \mathbf{0}$, then the right-hand inequality simplifies as follows:

$$f(\mathbf{x}^*) - \langle\, \mathbf{C}(\mathbf{x}^*), \boldsymbol{\lambda}^*\, \rangle \;\leq\; f(\mathbf{x}) - \langle\, \mathbf{C}(\mathbf{x}), \boldsymbol{\lambda}^*\, \rangle \tag{13a}$$

$$f(\mathbf{x}^*) - f(\mathbf{x}) \;\leq\; \langle\, \mathbf{C}(\mathbf{x}^*), \boldsymbol{\lambda}^*\, \rangle - \langle\, \mathbf{C}(\mathbf{x}), \boldsymbol{\lambda}^*\, \rangle \tag{13b}$$

$$f(\mathbf{x}^*) - f(\mathbf{x}) \;\leq\; -\langle\, \mathbf{C}(\mathbf{x}), \boldsymbol{\lambda}^*\, \rangle \tag{13c}$$

$$f(\mathbf{x}) - f(\mathbf{x}^*) \;\geq\; \langle\, \mathbf{C}(\mathbf{x}), \boldsymbol{\lambda}^*\, \rangle \tag{13d}$$

For all feasible $\mathbf{x}$, the condition $\mathbf{C}(\mathbf{x}) \geq \mathbf{0}$ holds, in which case the inner product on the right-hand side of (13d) is non-negative because both its operands are in the same orthant of $\mathbf{R}^m$.[7] This in turn implies that $f(\mathbf{x}^*)$ lies to the left of $f(\mathbf{x})$ on the real line *for all feasible* $\mathbf{x}$, i.e., $\mathbf{x}^*$ is a *global* minimiser of $f(\mathbf{x})$ ■

First, note that the theorem above assumes existence of the saddle point $(\boldsymbol{\lambda}^*, \mathbf{x}^*)$ but does not specify the form of f, only that it has to be minimized. The latitude afforded by this may be exploited to ensure existence of the saddle point $(\boldsymbol{\lambda}^*, \mathbf{x}^*)$. To that end, an obvious approach is to introduce a "link function" that couples the function f to the equation system $\mathbf{C}(\mathbf{x}) = \mathbf{0}$; and one may achieve such a linkage by defining f as a composite metric $d_c(\boldsymbol{\Phi}, \mathbf{0})$ that is a linear sum of the distance functions in equation (4), viz.:

$$f(\mathbf{x}) \equiv d_c(\boldsymbol{\Phi}, \mathbf{0}) = d_1(\boldsymbol{\Phi}, \mathbf{0}) + d_2(\boldsymbol{\Phi}, \mathbf{0}) + d_\infty(\boldsymbol{\Phi}, \mathbf{0}) \tag{14}$$

where: $\boldsymbol{\Phi}(\mathbf{x}, \boldsymbol{\lambda}) \equiv \text{vec}\{\, \phi(\lambda_i, c_i(\mathbf{x}))\, \}$;

ϕ — is an NCP-function;

$\boldsymbol{\lambda}$ — is an arbitrary non-negative and non-zero constant vector.

Secondly, note that one is also free to use any type of NCP-function in equation (14); but this paper assumes (and GENO employs) the Fischer-Burmeister function that was presented earlier.

[5] Similar sufficiency theorems but of a local nature and with different styles of proof may be found in [**14**, p.74] and [**29**, p. 74].

[6] This result also follows directly from the orthogonality condition, $\langle\, \boldsymbol{\lambda}, \mathbf{C}(\mathbf{x})\, \rangle = 0$, of the first-order optimality criteria.

[7] A simple proof of this assertion is as follows: in expanded form, the inner product function is simply a sum of corresponding elements of its vector operands, viz.: $\langle \mathbf{a}, \mathbf{b} \rangle = \sum a_i b_i$. If **a** and **b**, are in the same orthant, then corresponding non-zero elements have the same sign and their product is therefore non-negative; and the sum of non-negative products is obviously non-negative; if one of the vectors is in the orthant's interior and the other is non-zero, then by .the same argument, it should be apparent that the inner product is strictly positive.

Finally, note that $\boldsymbol{\lambda}^*$ is an arbitrary fixed vector which is pre-set; in other words, the maximization process implied by the left-hand inequality in (11b) is not necessary and so the determination of the saddle point is solely down to the minimization process implied by the right-hand inequality. Thus, the equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ may be solved via the following uni-objective optimization problem:

MP_{ec}: $\quad \underset{\mathbf{x}}{\text{Min}} \; L_e(\boldsymbol{\lambda}^*, \mathbf{x})$

Subject to: $c_i(\mathbf{x}) = 0; \quad x_j \in [L_j, U_j], \quad \forall i \in \{1, 2, \ldots, m\}; \; \forall j \in \{1, 2, \ldots, n\}$

Remarks 6. The definition of f in equation (14) is consistent with what constitutes a merit function for an NCP as defined by Facchinei & Soares [**5**]; in fact their own merit function — the inner product $\langle \boldsymbol{\Phi}, \boldsymbol{\Phi} \rangle$ which is now commonly known as 'natural merit function of an NCP' — is simply the square of the second term in equation (14). The model MP_{ec} is reliable in the sense that once a minimizer $\mathbf{x}^*$ of $L_e(\boldsymbol{\lambda}^*, \mathbf{x})$ is found, then we can rest assured that it is also a *global* minimizer f and a solution to the system $\mathbf{C}(\mathbf{x}) = \mathbf{0}$;[8] this is much unlike other merit function methods where no such guarantee can be made.[9] The challenge of course is to design algorithms that are capable of finding global optima in general.

7 Numerical Examples

This section presents the results obtained by GENO for some practical nonlinear equation systems presented in [**1**], [**9**], [**10**], [**13**], [**15**] and [**30**]. In order to limit the length of this paper, detailed numerical results are presented for only two problems that are mathematical models of the steady-state of real chemical processes: the first is an equation system describing the equilibrium products of a hydrocarbon combustion process; the second is an equation system describing the production of synthesis gas in an adiabatic reactor; results for the rest of the examples are reported only in comparative summary form in §7.3, but are available in full in [**24**].

7.1 Example 1: Hydrocarbon Combustion Process [13, 15]

Given: $R_1 = 10; \quad R_2 = 0.193; \quad R_3 = 4.106 * 10^{-4}; \quad R_4 = 5.451 * 10^{-4};$
$R_5 = 4.497 * 10^{-7}; \quad R_6 = 3.407 * 10^{-5}; \quad R_7 = 9.615 * 10^{-7}$

Solve the System:

$$x_1x_2 + x_1 - 3x_5 = 0 \tag{i}$$

$$2x_1x_2 + x_1 + x_2x_3^2 + R_5x_2 - R_1x_5 + 3R_7x_2^2 + R_4x_2x_3 + R_6x_2x_4 = 0 \tag{ii}$$

$$2x_2x_3^2 + 2R_2x_3^2 - 8x_5 + R_3x_3 + R_4x_2x_3 = 0 \tag{iii}$$

$$R_6x_2x_4 + 2x_4^2 - 4R_1x_5 = 0 \tag{iv}$$

$$x_1(x_2 + 1) + R_7x_2^2 + x_2x_3^2 + R_5x_2 + R_2x_3^2 + x_4^2 - 1 + R_3x_3 + R_4x_2x_3 + R_6x_2x_4 = 0 \tag{v}$$

Subject to: $x_i \in [0.00001, 100], \forall i \in \{1, 2, \ldots, 5\}.$

[8] This statement should be obvious from the very definition of $f(\mathbf{x})$ in (14), but a simple proof is as follows: for a given positive vector $\boldsymbol{\lambda}^*$, let $\mathbf{S}$ be the set of all global minimizers of $f(\mathbf{x})$; assume the vector equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ has a solution $\mathbf{u} \notin \mathbf{S}$; then $\forall i$, $c_i(\mathbf{u}) = 0$, and by definitions (10a) and (7), it follows that $\boldsymbol{\Phi}(\boldsymbol{\lambda}^*, \mathbf{u}) = \mathbf{0}$; this of course implies that $f(\mathbf{u}) = 0$ by definition (14); in other words, $\mathbf{u}$ is also a global minimizer of $f(\mathbf{x})$ which contradicts the assumption that $\mathbf{u}$ *is not in* $\mathbf{S}$. Therefore we must have that $\mathbf{u} \in \mathbf{S}$. Next, consider a typical element $\mathbf{v} \in \mathbf{S}$ and assume $\mathbf{C}(\mathbf{v}) \neq \mathbf{0}$; by (14), the statement $f(\mathbf{v}) = 0$ implies $\boldsymbol{\Phi}(\boldsymbol{\lambda}^*, \mathbf{v}) = \mathbf{0}$, i.e. $\forall i$, $c_i(\mathbf{v}) = 0$, which contradicts the assumption that $\mathbf{C}(\mathbf{v}) \neq \mathbf{0}$. But since $\mathbf{v}$ is arbitrary, one may assert that *all* elements of $\mathbf{S}$ are solutions to the vector equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$, and conversely, *all* solutions of the said equation are elements of $\mathbf{S}$.

[9] Nocedal & Wright [**19**, p. 286] cite the single variable example $f(x) = 0.5(\text{Sin}(5x) - x)^2$ which has seven local minima, but only three of which are also solutions to the equation $f(x) = 0$; merit function methods that try to find the roots of $f(x) = 0$ by solving the auxiliary program Min $\| f(x) \|_2$ may not yield the correct solution unless if the solution algorithm is started at a fortuitous point.

7.1.1 Variable Endogenization

A very useful technique that one may deploy on some problems prior to optimization is to "endogenize" some decision variables. At the very least, endogenization reduces the dimension of the search space which in turn often leads to a more efficient search. Though seemingly novel, the technique is in essence akin to the Gaussian elimination method that has been known for centuries [**8**]. Its application is described in more detail in [**25**]; here, it suffices to describe the algorithm in summary form as follows:

Step 1: Create a 'connexion matrix'—previously known as 'incidence matrix'— that indicates the presence or absence of each variable per equation, viz.:

EQUATION NO	x_1	x_2	x_3	x_4	x_5
i	✓	✓			✓
ii	✓	✓	✓	✓	✓
iii		✓	✓		✓
iv		✓		✓	✓
v	✓	✓	✓	✓	

Step 2: Identify equations to eliminate by "endogenizing" variables based on whether the chosen equation is easy to manipulate to make the chosen variable the subject of the equation. By this heuristic, one can see that equations (i), (ii) and (iii) as prime targets for elimination using variables x_1, x_4 and x_5 respectively.

Step 3: To ensure non-circular definitions, manipulate the connexion matrix into "row echelon" form (by swapping rows and / or columns) such that the variables identified for "endogenization" are on the "sloping edge", viz.:

EQUATION NO	x_2	x_3	x_5	x_1	x_4
iii	✓	✓	✓		
i	✓		✓	✓	
ii	✓	✓	✓	✓	✓
iv	✓		✓		✓
v	✓	✓		✓	✓

Step 4: Using items comprising the "slopping edge" of the echelon, i.e., equation (iii), (i) and (ii) and variables x_5, x_1 and x_4, define the endogenous equations, viz.:

$$z_1 \equiv x_5 = [2x_2x_3^2 + 2R_2x_3^2 + R_3x_3 + R_4x_2x_3]/8 \tag{15a}$$

$$z_2 \equiv x_1 = 3x_5/[x_2 + 1] \tag{15b}$$

$$z_3 \equiv x_4 = [R_1x_5 - 2x_1x_2 - x_1 - x_2x_3^2 - R_5x_2 - 2R_7x_2^2 - R_4x_2x_3]/R_6x_2 \tag{15c}$$

Step 5: For convenience, rename the variables as follows:

OLD NAME	x_5	x_1	x_4	x_2	x_3
Intermediate	z_1	z_2	z_3	-	-
NEW NAME	z_1	z_2	z_3	x_1	x_2

Step 6: Restate the equation system using the "new" variables from Step 5, viz.:

Given: $R_1 = 10$; $R_2 = 0.193$; $R_3 = 4.106 * 10^{-4}$; $R_4 = 5.451 * 10^{-4}$;
$R_5 = 4.497 * 10^{-7}$; $R_6 = 3.407 * 10^{-5}$; $R_7 = 9.615 * 10^{-7}$

$$z_1 \equiv [2x_1x_2^2 + 2R_2x_2^2 + R_3x_2 + R_4x_1x_2]/8$$

$$z_2 \equiv 3z_1/[x_1 + 1]$$

$$z_3 \equiv [R_1z_1 - 2z_2x_1 - z_2 - x_1x_2^2 - R_5x_1 - 3R_7x_1^2 - R_4x_1x_2]/R_6x_1$$

Solve the System: $R_6x_1z_3 + 2z_3^2 - 4R_1z_1 = 0$

$$z_2(x_1 + 1) + R_7x_1^2 + x_1x_2^2 + R_5x_1 + R_2x_2^2 + z_3^2 - 1 + R_3x_2 + R_4x_1x_2 + R_6x_1z_3 = 0$$

Subject to: $x_i \in [0.00001, 100], \forall i \in \{1, 2\}$; $z_i \in [0.00001, 100], \forall i \in \{1, 2, 3\}$.

7.1.2 Solution of Example 1 by Optimization Model MP_{em1}

Given: R_i, $i = 1, 2, 3, \ldots, 7$ as defined above;

$$z_1 \equiv [2x_1x_2^2 + 2R_2x_2^2 + R_3x_2 + R_4x_1x_2]/8$$

$$z_2 \equiv 3z_1/[x_1 + 1]$$

$$z_3 \equiv [R_1z_1 - 2z_2x_1 - z_2 - x_1x_2^2 - R_5x_1 - 3R_7x_1^2 - R_4x_1x_2]/R_6x_1$$

$$\underset{\mathbf{x}}{\text{Opt}} \{ |c_1(\mathbf{x})|, |c_2(\mathbf{x})| \}$$

Subject to: $c_1 \equiv R_6x_1z_3 + 2z_3^2 - 4R_1z_1 = 0$

$$c_2 \equiv z_2(x_1 + 1) + R_7x_1^2 + x_1x_2^2 + R_5x_1 + R_2x_2^2 + z_3^2 - 1 + R_3x_2 + R_4x_1x_2 + R_6x_1z_3 = 0$$

$x_i \in [0.00001, 100], \forall i \in \{1, 2\}$; $z_i \in [0.00001, 100], \forall i \in \{1, 2, 3\}$.

GENO Output

Generation	*Time*	c_1	c_2
0	0.02	25.63170487	11.87161090
10	10.13	0.00008544	0.99677931
20	10.44	0.00094296	0.00004669
30	10.42	0.00000019	0.00000003
40	10.52	0.00000000	0.00000000
50	10.58	0.00000000	0.00000000
60	10.53	0.00000000	0.00000000
70	10.61	0.00000000	0.00000000
80	10.47	0.00000000	0.00000000
90	10.41	0.00000000	0.00000000
100	10.42	0.00000000	0.00000000

Optimal Decision Vector: $\mathbf{x}^*$ = (34.165623853265423, 0.065451818001427)^T
Optimal Endogenous Vector: $\mathbf{z}^*$ = (0.036953303429454458, 0.0031525079933444459, 0.85939856226722866)^T
Optimal Equation Vector: $\mathbf{C}(\mathbf{x}^*)$ = (0.00000000, 0.00000000)^T
Average execution time per 10 generations: 10.45 seconds
Overall execution time on 100 generations: 104.52 seconds
Approximate time to first optimum:[10] 41.80 seconds

[10] This is an approximate measure of 'time performance' that is computed as follows: the solution first emerges at generation 40 and so the 'time-to-first-optimum' is approximately: 40 generations @ 10.45seconds per decade = 41.80 seconds.

7.1.3 Solution of Example 1 by Optimization Model MP_{ed}

Given: R_i, $i = 1, 2, 3, \ldots, 7$ as defined above;

$$z_1 \equiv [2x_1x_2^2 + 2R_2x_2^2 + R_3x_2 + R_4x_1x_2]/8$$

$$z_2 \equiv 3z_1/[x_1 + 1]$$

$$z_3 \equiv [R_1z_1 - 2z_2x_1 - z_2 - x_1x_2^2 - R_5x_1 - 3R_7x_1^2 - R_4x_1x_2]/R_6x_1$$

$$\min_{\mathbf{x}} \{ d_c(\mathbf{J},\mathbf{0}) \equiv d_1(\mathbf{J},\mathbf{0}) + d_2(\mathbf{J},\mathbf{0}) + d_\infty(\mathbf{J},\mathbf{0}) \}$$

Subject to: $c_1 \equiv R_6x_1z_3 + 2z_3^2 - 4R_1z_1 = 0$

$$c_2 \equiv z_2(x_1 + 1) + R_7x_1^2 + x_1x_2^2 + R_5x_1 + R_2x_2^2 + z_3^2 - 1 + R_3x_2 + R_4x_1x_2 + R_6x_1z_3 = 0$$

$x_i \in [0.00001, 100]$, $\forall i \in \{1, 2\}$; $z_i \in [0.00001, 100]$, $\forall i \in \{1, 2, 3\}$.

GENO Output

Generation	*Time*	d_c
0	0.00	241171.53574984
10	4.88	2.98435237
20	4.99	0.00090881
30	4.96	0.00000100
40	4.98	0.00000000
50	4.93	0.00000000
60	5.04	0.00000000
70	5.10	0.00000000
80	4.95	0.00000000
90	4.98	0.00000000
100	4.99	0.00000000

Optimal Decision Vector: $\mathbf{x}^*$ = (34.165623853267761, 0.065451818001432)^T

Optimal Endogenous Vector: $\mathbf{z}^*$ = (0.036953303429462597, 0.0031525079933449312, 0.8593985622675048 8)^T

Optimal Equation Vector: $\mathbf{C}(\mathbf{x}^*)$ = (0.00000000, 0.00000000)^T

Average execution time per 10 generations: 4.98 seconds

Overall execution time on 100 generations: 49.80 seconds

Approximate time to first optimum: 19.92 seconds

General Remarks

Equation systems emanating from chemical engineering tend to be very complex, with several possible types of multiplicity [**21**], and often rather sensitive in the sense that a unit change in (say) the 13th decimal place can drastically alter the outcome vector in some cases, hence the reporting of all the significant figures in $\mathbf{x}^*$ and $\mathbf{z}^*$ as generated by GENO. Also, recall that the final variable replacement table used at Step 5 of the 'endogenization' procedure is:

OLD NAME	x_5	x_1	x_4	x_2	x_3
Intermediate	z_1	z_2	z_3	-	-
NEW NAME	z_1	z_2	z_3	x_1	x_2

And so, in terms of the original variables, the optimal solution in this case is:

$\mathbf{x}^*$ = (0.0031525079933449312, 34.165623853267761, 0.065451818001432, 0.85939856226750488, 0.036953303429462597)^T

As expected, MP_{ed} exhibits a significant improvement in time performance compared to model MP_{em1}.

7.1.4 Solution of Example 1 by Optimization Model MP_{ec}

Given: R_i, $i = 1, 2, 3, \ldots, 7$ as defined above;

$$z_1 \equiv [2x_1x_2^2 + 2R_2x_2^2 + R_3x_2 + R_4x_1x_2]/8$$

$$z_2 \equiv 3z_1/[x_1 + 1]$$

$$z_3 \equiv [R_1z_1 - 2z_2x_1 - z_2 - x_1x_2^2 - R_5x_1 - 2R_7x_1^2 - R_4x_1x_2]/R_6x_1$$

$$\underset{\mathbf{x}}{\text{Min}}\ L_e(\boldsymbol{\lambda}, \mathbf{x})$$

Subject to: $c_1 \equiv R_6x_1z_3 + 2z_3^2 - 4R_1z_1 = 0$

$$c_2 \equiv z_2(x_1 + 1) + R_7x_1^2 + x_1x_2^2 + R_5x_1 + R_2x_2^2 + z_3^2 - 1 + R_3x_2 + R_4x_1x_2 + R_6x_1z_3 = 0$$

$x_i \in [0.00001, 100]$, $\forall i \in \{1, 2\}$; $z_i \in [0.00001, 100]$, $\forall i \in \{1, 2, 3\}$.

GENO Output

Generation	*Time*	L_e
0	0.00	99982.47570369
10	4.10	-0.75803267
20	4.21	-0.00055009
30	4.29	-0.00000019
40	4.34	0.00000000
50	4.09	0.00000000
60	4.06	0.00000000
70	4.09	0.00000000
80	4.07	0.00000000
90	4.12	0.00000000
100	4.10	0.00000000

Optimal Decision Vector: $\mathbf{x}^*$ = (34.165623853276642, 0.065451818001449)^T

Optimal Endogenous Vector: $\mathbf{z}^*$ = (0.036953303429491303, 0.0031525079933465839, 0.85939856226751088)^T

Optimal Equation Vector: $\mathbf{C}(\mathbf{x}^*)$ = (0.00000000, 0.00000000)^T

Average execution time per 10 generations: 4.15 seconds

Overall execution time on 100 generations: 41.47 seconds

Approximate time to first optimum: 16.60 seconds

General Remarks

The quality of a solution vector **x*** may be measured by how close each component of the original equation system is to zero when evaluated at **x***; Table 1 compares the GENO solution against those reported by others and, according to the 'success' criterion prescribed in §7.3, the GENO solution is best. The run-times for the EA-GA and PDA are 32.71 and 30 seconds respectively; Maranas & Floudas [**15**] report a time of 31.7 seconds

Table 1: A Comparative evaluation of "solutions" to **Example 1** computed by various methods

METHOD	EA-GA [9]	PDA [18]	Maranas & Floudas [15]	Kumar [13]	GENO
$f_1(\mathbf{x}^*)$	-0.1525772444	0.0036961619	-0.00000008	-0.00000236	0.00000000
$f_2(\mathbf{x}^*)$	-0.3712483541	-0.0036961549	0.00115049	-0.00000479	0.00099950
$f_3(\mathbf{x}^*)$	-0.0265535274	0.0036932686	-0.00000017	-0.00000031	0.00000000
$f_4(\mathbf{x}^*)$	-0.2784694038	0.0034008286	-0.00000014	0.00000193	0.00000000
$f_5(\mathbf{x}^*)$	-0.1168649340	-0.0007101592	-0.00000036	-0.00000275	0.00000000

7.2 Example 2: Production of Synthesis Gas in an Adiabatic Reactor [10, 13]

Solve the System:

$$x_1x_7 + 2x_2x_7 + x_3x_7 - 2x_6 = 0 \quad \text{(i)}$$

$$x_3x_7 + x_4x_7 + 2x_5x_7 - 2 = 0 \quad \text{(ii)}$$

$$7x_1 + 7x_2 + 7x_5 - 1 = 0 \quad \text{(iii)}$$

$$x_1 + x_2 + x_3 + x_4 + x_5 - 1 = 0 \quad \text{(iv)}$$

$$400x_1x_4^3 - 178370x_3x_5 = 0 \quad \text{(v)}$$

$$x_1x_3 - 2.6058x_2x_4 = 0 \quad \text{(vi)}$$

$$-28837x_1x_7 - 139009x_2x_7 - 78213x_3x_7 + 18927x_4x_7 + 8427x_5x_7 - 10690x_6 + 13492 = 0 \quad \text{(vii)}$$

Subject to: $x_i \in [0, 1], \forall i \in \{1, 2, \ldots, 5\}; \quad x_i \in [0, 5], \forall i \in \{6, 7\}.$

7.2.1 Preliminaries

Equations (iii) and (iv) together imply the following: $(7x_1 + 7x_2 + 7x_5 - 1 = 0) \wedge (7x_3 + 7x_4 - 6 = 0)$; and upon eliminating x_6 from (vii) using (i), one obtains the 'reduced' system shown below. The 'variable endogenization' technique of §7.1.1 was then applied on the 'reduced' system—the connexion matrix in "row echelon" form and the variable replacement table employed are as shown below.

Solve the System:

$$x_3x_7 + x_4x_7 + 2x_5x_7 - 2 = 0 \quad \text{(i)}$$

$$7x_1 + 7x_2 + 7x_5 - 1 = 0 \quad \text{(ii)}$$

$$7x_3 + 7x_4 - 6 = 0 \quad \text{(iii)}$$

$$400x_1x_4^3 - 178370x_3x_5 = 0 \quad \text{(iv)}$$

$$x_1x_3 - 2.6058x_2x_4 = 0 \quad \text{(v)}$$

$$-34182x_1x_7 - 149699x_2x_7 - 83558x_3x_7 + 18927x_4x_7 + 8427x_5x_7 + 13492 = 0 \quad \text{(vi)}$$

Subject to: $x_i \in [0, 1], \forall i \in \{1, 2, \ldots, 5\}; \quad x_i \in [0, 5], \ i \in \{7\}.$

Table 2a: Connexion Matrix in "Row Echelon" Form

EQUATION NO	x_1	x_4	x_5	x_3	x_2	x_7
iii		✓		✓		
ii	✓		✓		✓	
i		✓	✓	✓		✓
iv	✓	✓	✓	✓		
v	✓	✓		✓	✓	
vi	✓	✓	✓	✓	✓	✓

Table 2b: The Variable Replacement Table

OLD NAME	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Intermediate	-	z_2	z_1	-	-	-	z_3
NEW NAME	x_1	z_2	z_1	x_2	x_3	-	z_3

7.2.2 Solution of Example 2 by Optimization Model MP_{em1}

Given: $z_1 \equiv [6 - 7x_2]/7$

$z_2 \equiv [1 - 7x_1 - 7x_3]/7$

$z_3 = 2/[z_1 + x_2 + 2x_3]$

$$\underset{\mathbf{x}}{\text{Opt}} \left\{ |c_1(\mathbf{x})|, |c_2(\mathbf{x})|, |c_3(\mathbf{x})| \right\}$$

Subject to: $c_1 \equiv 400 x_1 x_2^3 - 178370 z_1 x_3 = 0$

$c_2 \equiv x_1 z_1 - 2.6058 x_2 z_2 = 0$

$c_3 \equiv -34182 x_1 z_3 - 149699 z_2 z_3 - 83558 z_1 z_3 + 18927 x_2 z_3 + 8427 x_3 z_3 + 13492 = 0$

$x_i \in [0, 1], \forall i \in \{1, 2, 3\};$

$z_i \in [0, 1], \forall i \in \{1, 2\}; \quad z_i \in [0, 5], i \in \{3\}.$

GENO Output

Generation	*Time*	c_1	c_2	c_3
0	0.00	158.42312143	0.01463718	31079.73906151
10	21.47	0.92121999	0.23190617	0.06239608
20	21.72	0.00008641	0.00002630	0.00008820
30	21.70	0.00000009	0.00000011	0.00000013
40	21.81	0.00000000	0.00000000	0.00000000
50	21.98	0.00000000	0.00000000	0.00000000
60	22.14	0.00000000	0.00000000	0.00000000
70	22.04	0.00000000	0.00000000	0.00000000
80	21.67	0.00000000	0.00000000	0.00000000
90	21.64	0.00000000	0.00000000	0.00000000
100	21.68	0.00000000	0.00000000	0.00000000

Optimal Decision Vector: $\mathbf{x}^*$ = (0.13110066819117702, 0.702222716807774040, 00065714291008900)^T

Optimal Endogenous Vector: $\mathbf{z}^*$ = (0.15492014033508308, 0.011099331755876839, 2.3297610327952265)^T

Optimal Equation Vector: $\mathbf{C}(\mathbf{x}^*)$ = (0.00000000, 0.00000000, 0.00000000)^T

Average execution time per 10 generations: 21.78 seconds

Overall execution time on 100 generations: 217.84 seconds

Approximate time to first optimum: 87.12 seconds

General Remarks

As mentioned previously, chemical engineering equations systems tend to be very sensitive—a unit change in (say) the 13th decimal place can drastically alter the outcome vector, hence the reporting of all significant figures in $\mathbf{x}^*$ and $\mathbf{z}^*$ as generated by GENO. The variable replacement table used at Step 5 of the 'endogenization' procedure is:

OLD NAME	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Intermediate	-	z_2	z_1	-	-	-	z_3
NEW NAME	x_1	z_2	z_1	x_2	x_3	-	z_3

And so, in terms of the original variables, the optimal solution in this case is:

$\mathbf{x}^*$ = (0.13110066819117702, 0.011099331755876839, 0.15492014033508308, 0.70222271680777404, 000657142910089, 0.359038857751804, 2.3297610327952265)^T

The optimal value for x_6 was calculated using the first equation of the original system; this procedure also applies to the solutions computed by MP_{em1} and MP_{ec}.

7.2.3 Solution of Example 2 by Optimization Model MP_{ed}

Given: $z_1 \equiv [6 - 7x_2]/7$

$z_2 \equiv [1 - 7x_1 - 7x_3]/7$

$z_3 = 2/[z_1 + x_2 + 2x_3]$

$$\underset{\mathbf{x}}{\text{Min}} \left\{ d_c(\mathbf{J},\mathbf{0}) \equiv d_1(\mathbf{J},\mathbf{0}) + d_2(\mathbf{J},\mathbf{0}) + d_\infty(\mathbf{J},\mathbf{0}) \right\}$$

Subject to: $c_1 \equiv 400 x_1 x_2^3 - 178370 z_1 x_3 = 0$

$c_2 \equiv x_1 z_1 - 2.6058 x_2 z_2 = 0$

$c_3 \equiv -34182 x_1 z_3 - 149699 z_2 z_3 - 83558 z_1 z_3 + 18927 x_2 z_3 + 8427 x_3 z_3 + 13492 = 0$

$x_i \in [0, 1], \forall i \in \{1, 2, 3\};$

$z_i \in [0, 1], \forall i \in \{1, 2\}; \quad z_i \in [0, 5], i \in \{3\}.$

GENO Output

Generation	*Time*	d_c
0	0.00	97665.55674122
10	2.71	4.03461584
20	2.81	0.77659804
30	2.82	0.67591134
40	2.86	0.64418526
50	2.94	0.45134032
60	3.03	0.06183755
70	2.87	0.00483354
80	2.89	0.00483354
90	2.89	0.00073943
100	2.92	0.00000000
110	2.82	0.00000000
120	2.79	0.00000000
130	2.79	0.00000000
140	3.00	0.00000000
150	2.87	0.00000000
160	2.95	0.00000000
170	2.92	0.00000000
180	2.93	0.00000000
190	2.89	0.00000000
200	2.89	0.00000000

Optimal Decision Vector: $\mathbf{x}^*$ = (0.13110066817468302, 0.702222716826246040, 0.0006571429101580)^T

Optimal Endogenous Vector: $\mathbf{z}^*$ = (0.15492014031661114, 0.011099331772301836, 2.3297610327948517)^T

Optimal Equation Vector: $\mathbf{C}(\mathbf{x}^*)$ = (0.00000000, 0.00000000, 0.00000000)^T

Average execution time per 10 generations: 2.88seconds

Overall execution time on 200 generations: 57.57 seconds

Approximate time to first optimum: 28.80 seconds

General Remarks

Although the algorithm was run for a longer period, MP_{ed} exhibits a significant improvement in time performance as compared to model MP_{em1}.

7.2.4 Solution of Example 2 by Optimization Model MP_{ec}

Given: $z_1 \equiv [6 - 7x_2]/7$

$z_2 \equiv [1 - 7x_1 - 7x_3]/7$

$z_3 = 2/[z_1 + x_2 + 2x_3]$

$$\underset{\mathbf{x}}{\text{Min}}\ L_e(\boldsymbol{\lambda}, \mathbf{x})$$

Subject to: $c_1 \equiv 400 x_1 x_2^3 - 178370 z_1 x_3 = 0$

$c_2 \equiv x_1 z_1 - 2.6058 x_2 z_2 = 0$

$c_3 \equiv -34182 x_1 z_3 - 149699 z_2 z_3 - 83558 z_1 z_3 + 18927 x_2 z_3 + 8427 x_3 z_3 + 13492 = 0$

$x_i \in [0, 1], \forall i \in \{1, 2, 3\};$

$z_i \in [0, 1], \forall i \in \{1, 2\}; \quad z_i \in [0, 5], i \in \{3\}.$

GENO Output

Generation	*Time*	L_e
0	0.00	-38180.11259736
10	2.95	0.61306758
20	2.96	0.00773979
30	2.95	0.00000194
40	2.98	0.00000000
50	2.96	0.00000000
60	2.95	0.00000000
70	2.96	0.00000000
80	2.96	0.00000000
90	2.96	0.00000000
100	3.01	0.00000000

Optimal Decision Vector: $\mathbf{x}^*$ = (0.131100668191177020, 0.702222716807774040, 0.00065714291008900)^T

Optimal Endogenous Vector: $\mathbf{z}^*$ = (0.154920140335083080, 0.011099331755876839, 2.32976103279522650)^T

Optimal Equation Vector: $\mathbf{C}(\mathbf{x}^*)$ = (0.00000000, 0.00000000, 0.00000000)^T

Average execution time per 10 generations: 2.97 seconds

Overall execution time on 100 generations: 29.66 seconds

Approximate time to first optimum: 11.88 seconds

General Remarks

A comparison in quality of the GENO solutions versus those reported by others is as shown below.

Table 3: A Comparative evaluation of "solutions" to **Example 2** computed by various methods

METHOD	$f_1(\mathbf{x}^*)$	$f_2(\mathbf{x}^*)$	$f_3(\mathbf{x}^*)$	$f_4(\mathbf{x}^*)$	$f_5(\mathbf{x}^*)$	$f_6(\mathbf{x}^*)$	$f_7(\mathbf{x}^*)$
Kumar [13]	7.76193E-06	-5.56232E-06	1.350672100	-0.00000270	0.000151201	4.23952E-08	0.073289564
Ji, *et al*. [10]	0.000000000	-7.10543E-15	1.350678638	0.00000000	3.81561E-12	-8.69096E-16	-1.70985E-10
GENO (MP_{em1})	0.000000000	-5.99520E-15	0.000000000	0.00000000	-1.74083E-13	-3.56459E-12	7.82165E-11
GENO (MP_{ed})	0.000000000	-6.43929E-15	0.000000000	0.00000000	-1.86873E-12	-2.02937E-12	8.91305E-11
GENO (MP_{ec})	0.000000000	-7.99361E-15	0.000000000	0.00000000	-7.49623E-13	9.19816E-13	1.01863E-10

7.3 Quality of Final Solution Comparison

The comparative analysis summarised by Table 4c is deliberately basic because a fuller study (whose protocol, in my opinion, ought to include the pre-optimization processing involved), is a major undertaking in itself. Thus, the only concern here is whether a method successfully converges onto at least one solution of the vector equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$. The quality of a candidate solution vector $\mathbf{x}^*$ may be measured by how close each component of the original equation system is to zero when evaluated at $\mathbf{x}^*$. Following Van Hentenryck, *et al.* [**30**], a candidate solution vector $\mathbf{x}^*$ was considered a 'successful solution' in this study if, under the mapping $\mathbf{C}(\cdot)$, all elements of the criterion set $\{|c_1|, |c_2|, \ldots, |c_m|\}$ were within the interval $[0, 10^{-8})$, i.e. when all components of the original vector equation are correct to the eighth decimal place.

The entries in Table 4c are two different measures of the quality of a solution $\mathbf{x}^*$ computed by a method on a given problem: (i) the integer in the top row indicates how many of the components c_i are outside the solution set $[0, 10^{-8})$; (ii) the real number in the bottom row is the actual value of the "offending" component that is furthest from the solution interval. Thus a method that 'successfully' finds the solution to a particular problem would have entries of '0' and '0.00000000' in the top and bottom rows against that problem; a method that fails by three components and with the worst of the "non-solutions" being -0.00045678 would have upper and lower entries of '3' and '0.00045678' respectively.

Brief descriptions of the methods considered as well as their sources are presented in Table 4a, and sources for the nonlinear vector equations involved are listed in Table 4b; a full set of results as computed by models MP_{em1}, MP_{ed} and MP_{ec} for each of the problems in Table 4b (and more) may be found in [**24**].

Table 4a: Primary and Secondary Sources for the Solution Techniques Compared

Algorithm #	Brief Description of Method and Source	Algorithm #	Brief Description of Method and Source
1	Newton's Method; see e.g., [**19**]	**6**	The Probability-driven Method in [**18**]
2	The Secant Method; see e.g., [**19**]	**7**	The Branch-and-Prune Method in [**30**]
3	Broyden's Method; see e.g., [**19**]	**8**	The Branch-and-Bound Method in [**15**]
4	The Optimal Time Method in [**2**]	**9**	The Branch-and-Bound Method in [**13**]
5	The Evolutionary Method in [**9**]	**10**	The Evolutionary Method GENO [**25**]

Table 4b: Some Secondary Sources for the Example Problems Utilised in the Evaluation

Problem #	Sources	Problem #	Sources
1	Example 6.2 in [**2**], or Example 1 in [**9**]	**7**	Benchmark i4 in §6.3 of [**30**]
2	Example 6.1 in [**2**], or Example 2 in [**9**]	**8**	See §6.8 of [**30**], or Problem 2 in [**18**]
3	Example 5 in [**15**], or Example 7 in [**13**]	**9**	See §6.4 of [**30**], or Problem 4 in [**18**]
4	Example 3 in [**15**], or Example 5 in [**13**]	**10**	See §6.6 of [**30**], or Problem 5 in [**18**]
5	Example 4 in [**15**]	**11**	See §6.5 of [**30**], or Problem 6 in [**18**]
6	Example 1 in [**15**], or Example 2 in [**13**]		

Table 4c: Quality of Final Solution Evaluation of Various Algorithms [11]

	Size	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4	Algorithm 5	Algorithm 6	Algorithm 7	Algorithm 8	Algorithm 9	GENO[12]
Problem 1	2	2 0.01496990	2 0.01496990	2 0.01496990	2 0.00738999	2 0.00126399	2 0.00000089				0 0.00000000
Problem 2	2				2 0.01922290	2 0.00276000	0 0.00000000				0 0.00000000
Problem 3	5								0 0.00000000	0 0.00000000	0 0.00000000
Problem 4	2								0 0.00000000	0 0.00000000	0 0.00000000
Problem 5	2								0 0.00000000	0 0.00000000	0 0.00000000
Problem 6	2								2 0.00229879	2 0.00000011	0 0.00000000
Problem 7	10					10 0.34472004	9 0.00000043	0 0.00000000			0 0.00000000
Problem 8	6					6 0.31396361	0 0.00000000	0 0.00000000			0 0.00000000
Problem 9	8					8 0.85265427	8 0.00000055	0 0.00000000			0 0.00000000
Problem 10	10					9 0.14824170	4 0.00000025	0 0.00000000			0 0.00000000
Problem 11	20					20 0.63991490	0 0.00000000	0 0.00000000			0 0.00000000

[11] The authors of Algorithm 8 and Algorithm 9 report their results in the decision space only; however, except Problem 6, their solution vectors **x*** coincide with those computed by GENO up to at least the eighth decimal place, and so one may safely assume their solutions in outcome space also meet the 'success' criteria prescribed in §7.3 above. But strictly speaking, the entries for Algorithm 7 should be regarded as tentative because the authors do not actually report the optimal decision vector but only that it is within intervals of width 10^{-8} or less; however, the intervals could be centred on the wrong value.

[12] Typical values for GENO's evolutionary parameters were as follows: probabilities for crossover operators – 0.55; mutation probability – 0.05; for Problems 1 - 6, the size of the mating population was 20; for Problems 7- 11, the mating population was either 20 or 30; the 'endogenization' technique (*supra*, pp. 12-13) was used on Problems 7, 9, 10 and 11.

8 Summary and Conclusions

It has been shown that one can always embed equation systems into a multi-objective optimization problem in which the criteria are the moduli of the functions c_i comprising the equation system, and the objective is to minimize the size of each $|c_i|$; the formulation labelled MP_{em1} accommodates both over-specified and under-specified equation systems.

The multi-objective formulation naturally suggests use of the Pareto-dominance notion in the quest for a solution. But, assuming the equation system is soluble, a decision-space/criterion-space examination of its solution shows it to be atypical of multi-objective problems: in criterion space the solution (known as the Pareto-set) is a singleton that also happens to be the ideal point; and in the decision space, the solution may be a singleton, a countable but finite set, or even an infinite set. The peculiar nature of the Pareto-set suggests that algorithms that rely solely on the Pareto-dominance notion may be found wanting in this case; though necessary, the Pareto-dominance condition may not be sufficient to ensure efficient convergence towards the Pareto set; what is required in addition is a mechanism that effectively "pulls" candidate solutions towards the ideal point, and the compromise solution concept embodies these twin mechanisms.

The basic multi-objective model is usually adequate on "small" equation systems ($m \leq 5$) but struggles to converge to the solution on larger systems. But the criterion vector $\mathbf{J} \equiv (|c_1|, |c_2|, \ldots, |c_m|)^T$ is in $\mathbf{R}^m$ — a space endowed with well known metrics based on the 'Euclidean', 'Tchebycheff' and 'Maximum' norms. And so, instead of the program 'Opt $\{ |c_1|, |c_2|, \ldots, |c_m| \}$', faster and higher quality performance may be achieved by minimizing the distance from the ideal point of the vector $\mathbf{J}$ as measured by the said metrics. Accordingly, two optimization models (labelled MP_{em2} and MP_{ed}) have been presented as alternatives to the basic model MP_{em1}; these are also be solved via the compromise solution concept.

A radically different approach called the NCP method has been proffered. This initially embeds the vector equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ into a nonlinear complementarity problem, and then into an auxiliary minimization problem of the form Min $\{f(\mathbf{x}) \mid \mathbf{C}(\mathbf{x}) \geq \mathbf{0}\}$. A sufficiency theorem shows that the solution to $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ may be characterized in terms of the saddle value of the Lagrangian $L_e(\boldsymbol{\lambda}, \mathbf{x}) \equiv f(\mathbf{x}) - \langle \boldsymbol{\lambda}, \mathbf{C}(\mathbf{x}) \rangle$, and it may be computed by simply minimizing L_e with respect to $\mathbf{x}$ and with $\boldsymbol{\lambda}$ fixed as an arbitrary positive vector. Out of the four models presented for solving the vector equation $\mathbf{C}(\mathbf{x}) = \mathbf{0}$, numerical results indicate this approach (labelled MP_{ec}) as being the most efficient method overall.

Although MP_{em1}, MP_{ed} and MP_{ec} produce only one solution, a comparative study summarised in Table 4c shows that, when judged purely on the quality of the said solution, the models' performances are very competitive—they match the branch-and-prune method [**30**] on Problems 7 - 11, but out-perform all other methods on all the examples problems presented. And in all cases, the endogenization technique proves to be useful—at the very least, it reduces the size of the search space which in turn often results in a more efficient solution process.

Although the comparative analysis presented in §7.3 is admittedly limited, one may still reasonably conclude that GENO partially disproves the hypothesis by Press, *et al*. [**20**, p.379] quoted at the beginning of this paper.

9 Legalities

I. Licence and Trademarks

II. Disclaimer

[13] Full URL: http://creativecommons.org/licenses/by-nc-sa/4.0/

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

References

1. Averick, B. M., Carter, R. G., More, J. J., Xue, G-L.: The MINPACK-2 test problem collection. *Technical Report ANL / MCS-TM-150*. Argonne National Laboratory, Argonne (1992).

2. Basirzadeh, H., Kamyad A. V., Effati, S.: An approach for solving a system of nonlinear equations in minimum time. *Indian J. Pure Appl. Math*., **34**, 947-961 (2003).

3. Coello Coello, C. A.: Treating constraints as objectives for single-objective evolutionary optimization. *Eng. Optim*., **32**, 275-308 (2000).

4. Dennis, J. E., Schnabel, R. B.: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs (1983).

5. Facchenei, F., Soares, J.: A new merit function for nonlinear complementary problems and a related algorithm. *SIAM J. Optim*., **7**, 225-247 (1997).

6. Fischer, A.: A special Newton-type optimization method. *Optimization*, **24**, 269-284 (1992).

7. Fukushima, M.: Merit functions for variational inequality and complementarity problems. In: Di Pillo, G., Giannessi, F. (eds.), *Nonlinear Optimization and Applications*, pp.155-170. Plenum, New York (1996).

8. Grcar, J. F.: How ordinary elimination became Gaussian elimination. *Historia Math*., **38**, 163-218 (2011).

9. Grosan, C., Abraham, A.: A new approach for solving nonlinear equations systems. *IEEE Trans. Syst, Man, Cybern. A, Syst., Humans*, **38**, 698-714 (2008).

10. Ji, Z., Wu, W., Li, Y., Feng, Y.: Numerical method for real root isolation of semi-algebraic system and its applications. To appear in *J. Comput. Appl. Math*. (2013).

11. Klamroth, K., Jørgen, T.: Constrained optimization using multiple objective programming. *J. Global Optim*., **37**, 325-355 (2007).

12. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In: Neyman, J. (ed.), *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492. University of California Press, Berkeley (1951).

13. Kumar, V.: *A Non-smooth exclusion test for finding all the solutions of nonlinear equations*. Unpublished M.Sc. Dissertation, MIT, Cambridge, Massachusetts (2007).

14. Mangasarian, O. L.: *Nonlinear Programming*. SIAM, Philadelphia (1994).

15. Maranas, C. D., Floudas, C. A.: Finding all solutions of nonlinearly constrained systems of equations. *J. Global Optim*., **7**, 143-182 (1995).

16. Marler, R. T., Arora, J. S.: Survey of multi-objective optimization methods in engineering. *Struct. Multidiscip. Optim*., **26**, 369-395 (2004).

17. Moore, R. E., Kearfott, R. B., Cloud, M. J.: *Introduction to Interval Analysis*. SIAM, Philadelphia (2009).

18. Nguyen Huu, T., Tran Van, H.: A new probabilistic algorithm for solving nonlinear equations systems. *Journal of Science, Special issue: Natural Sciences and Technology*, **30**, 1-17. Ho Chi Minh City: University of Education (2011).

19. Nocedal, J., Wright, S. J.: *Numerical Optimization*. Springer, New York (2006).

20. Press, W. H., Teukolsky, S. A., Vetterling W. T., Flannery, B. P.: *Numerical Recipes in* C*: The Art of Scientific Computing* (2nd ed.), Cambridge University Press, Cambridge (1992).

21. Rahimian, S. K., Jalai, F., Seader, J. D., White, R. E.: A new homotopy for seeking all real roots of a nonlinear equation. *Computers and Chemical Engineering*, **35**, 403-411 (2011).

22. Salukvadze, M. E.: Optimization of vector functionals. Part I: Programming of optimal trajectories (in Russian). *Avtomatika i Telemekhanika*, **8**, 5-15 (1971).

23. Siwale, I.: GENO™ 1.0: Supplement to User's Manual Part I: Static and Dynamic Programs. *Tech. Rep. RD-4-2005*, Apex Research Ltd, London (2005). [Online] Available at: http://www.researchgate.net

24. Siwale, I.: GENO™ 2.0: Supplement to User's Manual Part II: Nonlinear Equation Systems. *Tech. Rep. RD-19-2013*, Apex Research Ltd, London (2013). [Online] Available at: http://www.researchgate.net

25. Siwale, I.: GENO™ 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.

26. Siwale, I.: Practical multi-objective programming. *Tech. Rep. RD-14-2013*, Apex Research Ltd, London (2013). [Online] Available at: http://www.researchgate.net

27. Sun, D., Qi, L.: On NCP functions. *Comput. Optim. Appl*., **13**, 201-220 (1999).

28. Surry, P. D., Radcliffe, N. J., Boyd, I. D.: A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method. In: *Evolutionary Computing*, pp. 166-180. Springer Berlin Heidelberg (1995)

29. Takayama, A.: *Mathematical Economics*. Cambridge University Press, Cambridge (1985).

30. Van Hentenryck, P., McAllester, D., Kapur, D.: Solving polynomial systems using a branch and prune approach. *SIAM J. Numer. Anal*., **34**, 797-827, (1997).

31. Yu, P. L.: A class of solutions for group decision problems. *Management Sci*., **19**, 936-946 (1973).

32. Zeleny, M.: Compromise programming. In: Cochrane, J. L., Zeleny, M. (eds.), *Multiple Criteria Decision Making*, pp. 262-301. University of South Carolina Press, Columbia (1973).