Causality Assignment Using Multi-objective Evolutionary Algorithms

Tony Wong, Pascal Bigras, Karim Khayati
Department of automated production engineering
École de technologie supérieure, University of Québec
1100 rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3, Canada
E-mail: {tony.wong, pascal.bigras, karim.khayati}@etsmtl.ca

***Abstract*—Causality assignment is an important task in physical modeling by bond graphs. Traditional causality assignment procedures have specific aims and particular purposes. The traditional SCAP and MSCAP are algorithms that may fail if the bond graph has loops or contains junction causality violations. The RCAP algorithm focuses on the generation of differential algebraic equations to take into account junction violations caused by nonlinear multi-port devices and is not suitable for general bond graphs. We present a formulation of the causality assignment problem as a constrained multi-objective optimization problem. We then solve the resulting constrained system by a simple yet effective multi-objective evolutionary technique.**

Keywords: **Causality assignment, constrained optimization, evolutionary algorithms, multi-objective optimization, physical modeling**

I. INTRODUCTION

In physical modeling and simulation, system equations' formulation is generally based on the input-output relationships of all interconnected components. If the set of relationships is invariant then the task of equation formulation is trivial. However, topological invariance is seldom possible especially in cases where cross-domain applications are involved. For the latter cases, one has to resolve the proper input-output relationships before deriving system equations.

One approach to determine the input and output sets is to model system components as abstract multi-port devices. Then interconnect each device according to system topology and devices' intrinsic characteristics. Finally, perform a systematic analysis of the resulting device graph to yield useful information on the input and output sets. This approach corresponds to the use of bond graph as a framework to physical modeling and simulation. In bond graph formalism, the determination of the input and output sets is defined as a causality assignment procedure [1]. As it turns out, causality assignment not only determines the input and output sets but also detects the presence of algebraic loops and dependent storage elements. The knowledge of algebraic loops can provide better insight in the computer simulation of a given system. While the occurrence of dependent storage elements may indicate violations of principles of conservation of energy (i.e. nonlinear cross-domain system modeling or unintentional modeling errors).

This paper is organized as follows. In section II, we present an overview of the bond graph modeling technique with emphasis on the CAP (Causality Assignment Problem). The operating principles of PAES (Pareto Archived Evolution Strategy), a multi-objective optimization algorithm used to solve the CAP, are given in section III. Section IV details the mapping of the CAP into a constrained MOOP (Multi-Objective Optimization Problem) and its solution by the PAES. Finally, section V conatins some interesting results of this work.

II. BOND GRAPH MODELING

This section gives an overview of the bond graph formalism. It is based on the works by Karnopp et al. [1] and Van Dijk [2]. The goal of bond graphs is to represent a dynamic system by means of basic multi-port devices (or simply multi-ports) and their interconnections by bonds. These basic multi-ports exchange and modulate power through their bonds. There exist two power variables and two corresponding energy variables on each connected bond. They are: i) effort variable $e(t)$; ii) flow variable $i(t)$; iii) momentum $f(t)$; iv) displacement variable $q(t)$. The effort variable is the time derivative of some momentum and conversely, the momentum is the time integral of an effort,

$$e(t) = d\phi(t)/dt, \quad \phi(t) = \phi_0 + \int e(t)dt. \tag{1.1}$$

The same relationship applies to the flow variable and the displacement variable,

$$i(t) = dq(t)/dt, \quad q(t) = q_0 + \int i(t)dt. \tag{1.2}$$

Graphically a bond is a directed line connected to two ports sharing the bond variables. The direction of the line shows the power flow between ports and is called the power half arrow. The direction of the power flow is chosen by convention and does not need to reflect the true polarity of the flow. Fig. 1 illustrates the power flow between two multi-ports m_1 and m_2.

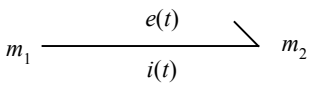

Fig. 1. A bond connecting two multi-ports m_1 and m_2. The half arrow indicates that power flows from m_1 to m_2.

A. Multi-ports

The basic multi-ports of a bond graph are elements that are source, dissipator, storage and modulator of power [2]. Table I enumerates the set of basic multi-ports. Note that in bond graph modeling it is not necessary to have a bijective mapping between the basic multi-ports and the system components. In fact, most of the basic multi-ports can represent subsystems comprising a number of system

components. The important factor to consider is that each system component or subsystem obeys the constitutive laws of a given multi-port. In Table I, the second column presents the usual power flow convention of the multi-ports. The third column presents the mandatory, constrained, preferred and indifferent computational causality of the multi-ports. The fourth column presents the constitutive laws of the basic multi-ports.

TABLE I
BOND GRAPH'S BASIC MULTI-PORTS ELEMENTS

Multi-port Name	Power flow	Computational Causality	Constitutive laws
Effort source	S_e	S_e	$e(t) = E(t)$
Flow source	S_f	S_f	$i(t) = I(t)$
Resistor	R	R R	$e(t) = f_R i(t)$, $i(t) = f_R^{-1} e(t)$.
Capacitor	C	C C	$e(t) = f_C^{-1} \int i(t)dt$, $i(t) = f_C \, de(t)/dt$.
Inertia	L	L L	$i(t) = f_L^{-1} \int e(t)dt$, $e(t) = f_L \, di(t)/dt$.
Transformer	1 — m $\ddot{TF}$ — 2	1 — m $\ddot{TF}$ — 2 1 — m $\ddot{TF}$ — 2	$e_1 = m\, e_2$, $i_2 = m\, i_1$. $i_1 = m^{-1} i_2$, $e_2 = m^{-1} e_1$.
Gyrator	1 — r $\ddot{GY}$ — 2	1 — r $\ddot{GY}$ — 2 1 — r $\ddot{GY}$ — 2	$e_1 = r\, i_2$, $e_2 = r\, i_1$. $i_1 = r^{-1} e_2$, $i_2 = r^{-1} e_1$.
0-junction	1 — n 0 n-1	1 — n 0 n-1	$e_1 = \ldots = e_{n-1} = e_n$, $i_1 + \cdots + i_{n-1} + i_n = 0$.
1-junction	1 — n 1 n-1	1 — n 1 n-1	$i_1 = \ldots = i_{n-1} = i_n$, $e_1 + \cdots + e_{n-1} + e_n = 0$.

The small stroke, called the causal stroke, at one end of a bond indicates the direction of travel of the effort variable information. The reaction to the effort information is the presence of a flow variable traveling in the opposite direction. Thus, the causal stroke represents the flow causality at one end of a bond. The opposite end of a bond must have complementary causality. This constitutes the fundamental causal constraint of a bond graph. Fig. 2 shows the presence of a causal stroke which determines the direction of travel of $e(t)$ and $i(t)$.

The 1-port sources S_e and S_f represent the interaction of a system with its environment. For example, in electrical and electronics systems, they may represent voltage and current sources.

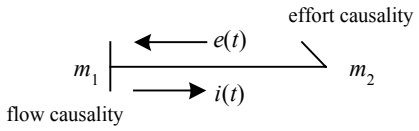

Fig. 2. The fundamental causal constraint: Causality must be complementary at both ends of a bond.

The 1-port resistive element R, capacitor element C and inertia element L behave analogously to their electrical counterpart. The 2-port transformer and gyrator are power continuous elements (no power storage and no power dissipation). The n-port junction elements are also power continuous. The constitutive laws of a 0-junction are analogous to the Kirchoff's current law. The constitutive laws of a 1-junction are analogous to the Kirchoff's voltage law.

B. Causality Assignment

In bond graphs, the inputs and outputs are characterized by the effort causality and flow causality. Thus, causality assignment is a process by which the bond variables ($e(t)$ and $i(t)$) are partitioned into input and output sets. It establishes the cause and effect relationships between the factors of power. There exist four types of causal constraints in bond graphs:

- *Mandatory causality*
  The constitutive laws allow only one of the two port variables to be the output. Sources S_e and S_f have mandatory causality. Multi-port elements R, L, C, TF and GY can also have mandatory causality if their constitutive laws cannot be inverted.
- *Preferred causality*
  For the storage elements C and L, there can be time derivative causality or time integral causality. The preferred causality here refers to the integral causality of these elements.
- *Constrained causality*
  For TF, GY, 0- and 1-junction there are relations between the causality of the different ports of the element. The relations are causal constraints because the causality of a particular port imposes the causality of the other ports.
- *Indifferent causality*
  Indifferent causality means there is no causal constraints. The linear resistor element R exhibits indifferent causality since both power variables $e(t)$, $i(t)$ can be made member of the input and output sets.

Most traditional causality assignment procedures use a local constraint propagation scheme to label bond causality. From a starting point, usually one of the source elements, bond causality is assigned sequentially, according to the multi-port connected to the bond, until all element ports are labeled. These causality assignment procedures must also satisfy the four causality types and the fundamental causal constraint.

The SCAP [1] (Sequential Causality Assignment Procedure) is an example of causal labeling by local propagation. However, this widely used procedure may produce incorrect causality assignment if the bond graph contains loops [2].

The modified form of SCAP, called MSCAP, can resolve incorrect causality assignment and guarantee optimal labeling by introducing the so-called basis-variable order and some graph-theoretic considerations [2]. The MSCAP is still a local propagation scheme and is actually composed of two different sub-procedures. In order to decide which of the sub-procedures is applicable, it requires some non trivial preprocessing to determine: i) if the bond graph has one or more loops along some subgraphs comprising 0-, 1-junctions and transformer multi-ports; ii) if the loop gain of all causal cycles is different than +1. However, MSCAP may still fail if the bond graph has unsatisfiable causal constraints.

Bond graphs with unsatisfiable causal constraints are usually found in nonlinear applications. Most often their constitutive laws are not amendable to the invert operation. This results in mandatory causality that may lead to junction causality violations. The RCAP [3] (Relaxed Causality Assignment Procedure) is a method that assigns causality consistent to the nonlinear constitutive laws of the elements. The causality constraints for 0-, 1-junctions are not strictly enforced. However, the constitutive laws for the junctions are still maintained and are used to generate the so-called algebraic constraint equations [4]. The resulting system equations are known as DAEs (Differential Algebraic Equations). It is obvious that RCAP is useful when there are constraint violations caused by nonlinear elements. There exists also a modified form of RCAP and is designed to generate more efficient DAEs [2].

In summary, the traditional SCAP may produce incorrect assignment due to bond graph loops. The modified SCAP requires non-trivial preprocessing and may fail in presence of junction causality violations. RCAP and its modified form are mainly concerned with the generation of DAEs for nonlinear systems. They cannot detect unintentional modeling errors. In the following sections, we present an alternate formulation of the CAP that does not require a priori knowledge of the bond graph topology or its intent. This alternate formulation considers the CAP as a constrained MOOP and we apply a multi-objective evolutionary optimizer to solve the resulting constrained system.

III. MULTI-OBJECTIVE OPTIMIZATION

A MOOP (Multi-Objective Optimization Problem) has a number of objective functions which are to be minimized or maximized. In multi-objective optimization with conflicting objectives there is no single optimal solution. Instead a set of solutions exist which are all optimal with respect to some objectives [5]. This solution set arises because of trade-offs between conflicting objectives.

In this work, we apply an algorithm called PAES [6] (Pareto Archived Evolution Strategy) to solve the MOOP representing the CAP (Causality Assignment Problem). The PAES is based on the dominance concept and the resulting solution set contains non-dominated solutions. The set of non-dominated solutions is known as the Pareto-optimal set if it is the non-dominated set of the entire feasible search space. Assuming a minimization problem, the dominance relation is defined as follows [9]:

Definition 1. A vector $\boldsymbol{u} = [u_1, \ldots, u_K]$ is said to dominate a vector $\boldsymbol{v} = [v_1, \ldots, v_K]$ if both conditions i and ii are true:

i) The vector $\boldsymbol{u}$ is not worse than $\boldsymbol{v}$,

$$u_i \le v_i, \quad \forall i \in \{1,\ldots,K\}. \tag{2.1}$$

ii) The vector $\boldsymbol{u}$ is strictly better than $\boldsymbol{v}$ in at least one element,

$$\exists i \in \{1,\ldots K\} : u_i < v_i. \tag{2.2}$$

Note that the dominance relation is non reflexive, non symmetric but is transitive. All Pareto-based MOEA (Multi-Objective Evolutionary Algorithm) will attempt to find non-dominated solutions that are close to or are members of the Pareto-optimal set.

A. PAES – Pareto Archived Evolution Strategy

This technique was devised by Knowles and Corne as a simple yet efficient evolutionary algorithm for solving real world multi-objective optimization problems [6], [7], [8]. The basic PAES is a (1+1)-ES and uses only mutation on a single parent solution to create a single offspring. The following steps show the general operating principles of this algorithm.

i) Generate a random solution vector $\boldsymbol{x}$.
ii) Evaluate objectives' value based on $\boldsymbol{x}$ and add $\boldsymbol{x}$ to archive.
iii) Terminate algorithm if stopping criteria are satisfied. Otherwise continue to step iv).
iv) Mutate $\boldsymbol{x}$ (the parent) to produce offspring solution vector $\boldsymbol{c}$.
v) Evaluate objectives' value based on $\boldsymbol{c}$.
vi) if $\boldsymbol{c}$ is dominated by $\boldsymbol{x}$ then go to step iii).
vii) Compare offspring solution vector $\boldsymbol{c}$ with members of the archive. Update archive accordingly.
viii) Select new parent solution vector $\boldsymbol{x}$ then go to iii).

The PAES maintains a fixed-size archive of the best solutions found during the search. Initially the archive is empty. The mutation operation involves a gaussian probability distribution. There are three possible scenarios when comparing a parent solution vector $\boldsymbol{x}$ to the offspring solution vector $\boldsymbol{c}$. They are:

1. If $\boldsymbol{x}$ dominates $\boldsymbol{c}$, the offspring solution vector $\boldsymbol{c}$ is rejected. A new offspring will be created by mutating $\boldsymbol{x}$ for further processing.
2. If $\boldsymbol{c}$ dominates $\boldsymbol{x}$. The offspring solution $\boldsymbol{c}$ is accepted and will become the parent of the next iteration. A copy of the offspring $\boldsymbol{c}$ is also kept in the archive.
3. If both $\boldsymbol{x}$ and $\boldsymbol{c}$ are non-dominating each other. The offspring $\boldsymbol{c}$ will be compared to the members of the archive:
   a) If $\boldsymbol{c}$ is dominated by a member of the archive. The offspring solution vector $\boldsymbol{c}$ is rejected. A new offspring will be created by mutating $\boldsymbol{x}$ for further processing.
   b) If $\boldsymbol{c}$ dominates some members of the archive. The dominated members are removed from the archive. The offspring solution $\boldsymbol{c}$ is accepted and will

become the parent of the next iteration. A copy of the offspring ***c*** is also kept in the archive.

c) If ***c*** is not dominated by any member of the archive and ***c*** does not dominate any member of the archive, it is added to the archive only if there is an empty slot in the archive.

Note that in step 3c) the offspring solution vector ***c*** does not necessary become the parent solution of the next iteration. Since the parent of ***c*** is also in the archive. The PAES make use of a neighborhood density measure to decide which of the two solution vectors will become the parent of the next iteration. That is, the one residing in the least crowded area of the search space will become the parent. This neighborhood density measure is also used to determine who remains in the archive when it is full.

The neighborhood density measure in PAES is defined as the number of solution vectors residing in a particular *K*-dimensional hypercube with *K* being the number of objectives. Thus, in PAES, each objective is divided into 2^n equal divisions. The parameter *n* is a user-defined parameter called the depth value. Every solution vector in the fixed-size archive belongs to one of the 2^{nK} hypercubes. The goal of this partitioning scheme and the neighborhood density measure is to maintain diversity within the non-dominated solution archive. And by favoring the less crowded area of the archive, the PAES can also produce spreading of the non-dominated solutions along the Pareto-optimal front.

In summary, the PAES is a (1+1)-ES designed for the search of non-dominated solutions in an iterative manner. The PAES maintains an archive of the best non-dominated solutions. This archive is partitioned into 2^{nK} hypercubes to help maintain diversity of the archived solutions. The depth value *n* and the size of the archive are the only user-defined parameters of this algorithm.

IV. PROBLEM MAPPING

This section presents the mapping of the CAP into a constrained MOOP. The first step is to define the proper bond graph representation. Then we proceed to define the proper candidate solution representation. The other steps of the problem mapping procedure are given in the following subsections.

We define a bond graph as $BG = <G, I>$. *G* is a directed and labeled graph and *I* a function identifying the multi-port type. We represent the graph *G* as a triplet $G = (M, B, \lambda_m)$, where $M = \{m_1, \ldots, m_{\|M\|}\}$ is the set of multi-ports, $B = \{b_1, \ldots, b_{\|B\|}\}$ is the set of bonds and $\| M \|$ denotes the cardinality of the set *M*. For each bond $b \in B$, there is a ordered couple (m_i, m_j) joined by *b* which defines the power direction of the bond. While λ_m is the set of bonds incident to a multi-port $m \in M$. We define $\lambda^+ : B \to M$, a function that returns the starting multi-port of a bond. Similarly, we define $\lambda^- : B \to M$ as a function returning the ending multi-port of a bond. We thus have

$$\lambda_m = \left\{ b \mid b \in B, \lambda^+(b) = m \right\} \cup \left\{ b \mid b \in B, \lambda^-(b) = m \right\}. \quad (3)$$

Finally, the identifying function *I* of the bond graph is simply $I : M \to T$, where $T \in$ {SF, SF, R, L, TF, GY, 0-junction, 1-junction} are the basic multi-port element labels.

A. Candidate solution vector representation

The natural candidate solution representation is a $1 \times \|B\|$ vector. Each element of the vector represents a bond within the bond graph. Since every bond has two causal labels (one for each end) and are complementary because of the fundamental causal constraint, a binary-valued representation (0-value representing flow causality and 1-value representing effort causality or vice versa) is adequate to capture all causality assignment of a bond graph. More formally, let $L_b^m \in \{0,1\}$ be the value of an element, of a solution vector, corresponding to bond *b* connected to multi-port *m*. The fundamental causal constraint states that,

$$L_b^{m_i} = 1 - L_b^{m_j}, \quad \begin{matrix} m_i = \lambda^+(b), m_j = \lambda^-(b), \\ \forall b \in B. \end{matrix} \quad (4)$$

Thus a candidate solution is a vector where each element represents the causality at one end of a bond. The causality at the other end of a bond can easily obtained by (4). The archive maintains a population of non-dominated solution vectors and we have a population matrix defined by

$$\mathbf{X} = \left[x_{i,j} \right], \quad i = 1,2,\ldots,N \quad j = 1,2,\ldots,\|B\|; \quad (5)$$
$$x_{i,j} \in \{0,1\},$$

where *N* is the archive size. This binary encoding ensures that all candidate solutions always satisfy the fundamental constraint (4).

B. Objective functions evaluation

The main objective of the CAP is to satisfy, if possible, all mandatory, preferred and constrained causality of bond graph element ports while maintaining the fundamental causal constraint of the bond graph. In order to solve this problem by PEAS, it is necessary to have a measure that indicates how good the assignment was for a candidate solution vector. In this context, the evaluation of a candidate solution vector is simply the direct counting of the number of causal violations for each multi-port of the bond graph. Recall that a solution vector has binary-valued elements. Using the convention: a 0-value indicating flow causality and a 1-value indicating effort causality, it is trivial to derive the following counting schemes.

- Multi-port S_f, *L*

$$f_{S_f,L} = \sum_{b \in \lambda_m} L_b^m, \quad \forall m \in \left\{ M_{S_f} \right\} \cup \left\{ M_L \right\}, \quad (6.1)$$

where $M_{S_f} = \{ p \mid I(p) = \mathrm{SF}, \forall p \in M \}$ is the set of flow sources and $M_L = \{ p \mid I(p) = \mathrm{L}, \forall p \in M \}$ is the set of inertia elements in the bond graph.

- Multi-port S_e, *C*

$$f_{S_e,C} = \sum_{b \in \lambda_m} \left(1 - L_b^m \right), \quad \forall m \in \left\{ M_{S_e} \right\} \cup \left\{ M_C \right\}. \quad (6.2)$$

- Multi-port *TF*

$$f_{TF} = \sum_{b_i, b_j \in \lambda_m, i \neq j} \left| L_{b_i}^m + L_{b_j}^m - 1 \right|, \quad \forall m \in \left\{ M_{TF} \right\}, \quad (6.3)$$

where $\left| \cdot \right|$ denotes the absolute value operator.

- Multi-port *GY*

$$f_{GY} = \sum_{b_i, b_j \in \lambda_m, i \neq j} \left| L_{b_i}^m - L_{b_j}^m \right|, \quad \forall m \in \{M_{GY}\}. \tag{6.4}$$

- Multi-port 0-junction

$$f_{0\text{-junction}} = \left| \|\lambda_m\| - \sum_{\forall b \in \lambda_m} L_b^m - 1 \right|, \quad \forall m \in \{M_{0\text{-junction}}\}. \tag{6.5}$$

where $\|\cdot\|$ denotes the set cardinality.

- Multi-port 1-junction

$$f_{1\text{-junction}} = \left| \sum_{\forall b \in \lambda_m} L_b^m - 1 \right|, \quad \forall m \in \{M_{1\text{-junction}}\}. \tag{6.6}$$

Note that these counting schemes do not involve the linear resistor elements because they have indifferent causality. For completeness sake, the counting scheme for nonlinear resistors is given by (6.7).

- Multi-port nonlinear *R*

$$f_{R_x} = \sum_{\forall b \in \lambda_m} \left(L_b^m - 2x L_b^m + x \right), \quad \forall m \in \{M_{R_x}\}, \tag{6.7}$$

where $x \in \{0, 1\}$ and represents the mandatory causality imposed by the nonlinear constitutive laws.

C. Constrained Multi-Objective Optimization

The causality assignment problem can be expressed as a constrained MOOP given by:

$$\begin{aligned} \text{minimize} \quad & f_k(\mathbf{x}), \quad k = 1, \ldots, K; \\ s.t. \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, J; \\ & x_i \in \{0,1\}, \quad i = 1, \ldots, \|B\|. \end{aligned} \tag{7}$$

In this work, the *K* objective functions are the number of causal violations for the inertia, capacitor, 0- and 1-junction multi-ports. The *J* inequality constraints are the number of causal violations for the sources, *TF*, *GY* and nonlinear *R* multi-ports. All decision variables are binary-valued and their number is equal to that of the bonds. Thus, we have 4 objective functions represented by (6.1), (6.2), (6.5) and (6.6) and five inequality constraints represented by (6.1), (6.2), (6.3), (6.4) and (6.7). In this forumaltion, we try to enforce mandatory, constrained causality while minimizing non preferred causality and junction violations.

The standard PAES is an unconstrained MOEA. In order to solve (7), we need to adopt a suitable constraint handling technique. A number of constraint handling techniques are given in [10] and [11]. The penalty approach is the most straightforward *external* technique for the PAES. In this approach all inequality constraints are normalized (necessary when constraints are incommensurable). For each candidate solution vector we calculate the number of constraint violations using

$$\omega_j(\mathbf{x}) = \begin{cases} g_j(\mathbf{x}), & \text{if } g_j(\mathbf{x}) > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Then we sum all constraint violations from (8)

$$\Omega(\mathbf{x}) = \sum_{j=1}^{J} \omega_j(\mathbf{x}) \tag{9}$$

and add the resulting value of (9) to each of the *K* objective function value. Thus, we have

$$F_k(\mathbf{x}) = f_k(\mathbf{x}) + \alpha_k \Omega(\mathbf{x}), \quad k = 1, \ldots, K \tag{10}$$

where α_k are constants such that both terms on the right-hand side of (10) have the same magnitude. In this work, $\alpha_k = 1$, $\forall k$. For our PAES implementation, instead of direct evaluation of the objective functions, we will use (8) – (10) to take into account constraint violations.

D. Archive size and depth value

The archive size and depth value are two user-definable parameters of the PAES. The archive serves as a memory of non-dominated solutions found during execution of the algorithm. It is also used in the dominance ranking of the solution vectors. The archive size must not be too small otherwise the dominance ranking will suffer because of under sampling. The depth value defines the hypercube size (see section III). Note that it is difficult to control the diversity of the non-dominated solutions when the hypercubes are large. In order to partition the archive into 2^{nK} hypercubes, it is necessary to divide each objective into 2^n divisions. Therefore, we need to estimate the minimun and maximum values of the *K* objective functions. In this work, we have four objective functions, their bounding values are given below:

- $f_1 \equiv f_C$:

$$\min f_C = 0, \quad \max f_C = \|M_C\|. \tag{11.1}$$

- $f_2 \equiv f_L$:

$$\min f_L = 0, \quad \max f_L = \|M_L\|. \tag{11.2}$$

- $f_3 \equiv f_{0\text{-junction}}$:

$$\min f_{0\text{-junction}} = 0, \quad \max f_{0\text{-junction}} = \|M_{0\text{-junction}}\| \left(\|\lambda_m\| - 1 \right). \tag{11.3}$$

- $f_4 \equiv f_{1\text{-junction}}$:

$$\min f_{1\text{-junction}} = 0, \quad \max f_{1\text{-junction}} = \|M_{1\text{-junction}}\| \, \|\lambda_m\|. \tag{11.4}$$

Equations (11.1) to (11.4) give the minimum and maximum objective function values without the added penalty weight (10). Some solution vectors may have greater objective values because of constraint violations. These solution vectors will not be accepted into the archive.

V. RESULTS

This section presents some results using the PAES. As an example, we applied the PAES to a bond graph inspired by the hedgetrimmer system modeling study of [12]. Fig. 3 shows the bond graph representation of this system. Our hedgetrimmer model has 21 bonds, four nonlinear multi-ports. The NL1 box represents a subsystem that translates rotary displacement into linear displacement. The NL2 box converts the linear displacement into a nonlinear load to simulate the cutting action on branches and leaves.

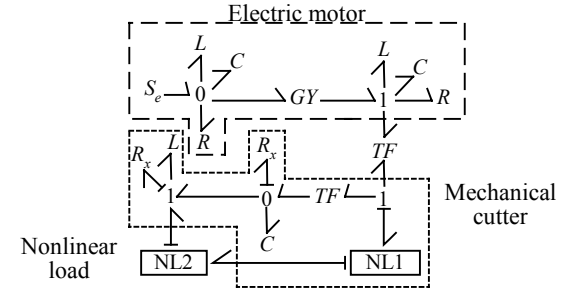

Fig. 3. Bond graph with nonlinear multi-ports and mandatory causality.

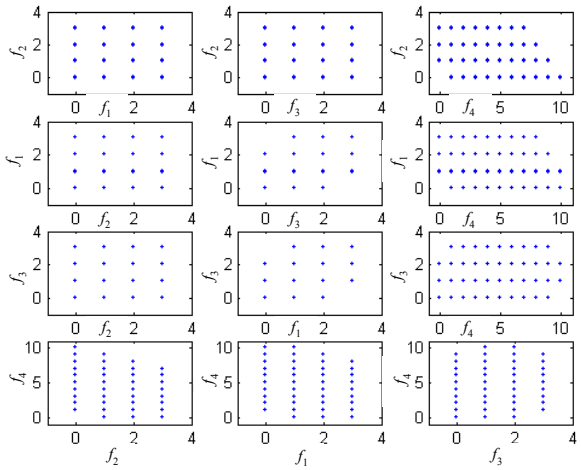

Fig. 4. Objective space of the causality assignment problem.

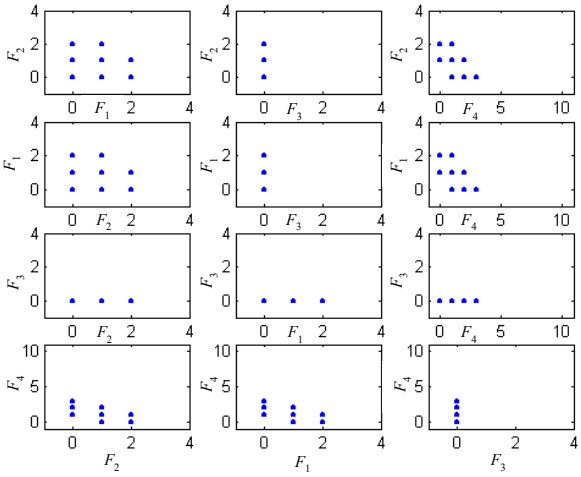

Fig. 5. Objective space of the non-dominated solutions obtained by PAES.

TABLE II
NON-DOMINATED SET OBJECTIVE FUNCTION VALUES

F_1	F_2	F_3	F_4	F_1	F_2	F_3	F_4
0	0	0	3	1	2	0	0
1	1	0	1	2	0	0	1
1	0	0	2	0	1	0	2
0	2	0	1	2	1	0	0

In Fig. 4, the pair-wise XY plots show the discrete objective space of the CAP. We applied the PAES with an archive size of 20 and a depth value of 3 (a total of 4096 hypercubes) to the problem. The stopping criterion was 1000 iterations of the algorithm. The resulting non-dominated solutions are plotted in Fig. 5 using their objective values. The same data are given in Table II in numerical form.

To select a unique solution from the non-dominated set, we first define an ideal vector f^* representing our preference in the objective space. Using an appropriate distance metric we then select the closest objective vector and its corresponding solution vector as the preferred solution of the problem. If we define $f^* = [0\ 0\ 0\ 0]^T$ and using the L_2-norm, the objective vector $[1\ 1\ 0\ 1]^T$ is the selected one. This vector corresponds to 1 non-preferred causality for C element, 1 non-preferred causality for L element and 1 junction violation. However, if we define $f^* = [2\ 2\ 0\ 0]^T$, the selected objective vector becomes $[1\ 2\ 0\ 0]^T$. That is, a causality assignment with 3 non preferred causality and no junction violation.

VI. REFERENCES

[1] J. Van Dijk, "On the role of bond graph causality in modeling mechatronics systems," Ph.D. dissertation, Univ. of Twente, Enschede, The Netherlands, 1994.

[2] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: A Unified Approach*, 2nd ed., New York: John Wiley & Sons, 1990.

[3] J. Van Dijk and P. C. Breedveld, "Relaxed causality: a bond graph oriented perspective on dae-modeling," *Proc. 1995 International Conf. on Bond Graph Modeling and Simulation,* pp. 225-231.

[4] B. J. Joseph and H. R. Martens, "The method of relaxed causality in the bond graph analysis of nonlinear systems," *ASME Trans. Journal of Dynamic Systems, Measurement and Control,* no. 96, pp. 95-99, 1974.

[5] A. Ben-Tal, "Characterization of Pareto and lexicographic optimal solutions," in *Multiple Objective Decision Making – Methods and Applications*, vol. 177, *Lecture Notes on Economics and Mathematical Systems*. Berlin: Springer-Verlag, 1980, pp. 1-11.

[6] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the Pareto archived evolutionary strategy," *Evolutionary Computation Journal,* vol. 8, no. 2, pp. 149-172.

[7] J. D. Knowles and D.W. Corne, "The Pareto Archived Evolution Strategy : A New Baseline Algorithm for Pareto Multiobjective Optimisation," in *Proceedings of the 1999 Congress on Evolutionary Computation*, pp. 98-105.

[8] J. D. Knowles and D.W. Corne, "Local Search, Multiobjective Optimization and the Pareto Archived Evolution Strategy," in *Proceedings of the Third Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, pp. 209-216.

[9] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part I: A unified formulation," *IEEE Transactions on Systems, Man and Cybernatics*, vol. 28, no. 1, 1998, pp. 26-37.

[10] C. A Coello, "Constraint-handling using an evolutionary multiobjective optimization technique," *Civil Engineering Systems*, Gordon and Breach Science Publishers, vol. 17, 2000, pp. 319-346.

[11] C. A. Coello, "Constraint handling through a multiobjective optimization technique," in Annie S. Wu, editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, pp. 117-118.

[12] M. A. Atherton, R. A. Bates, "Bond graph analysis in robust engineering design," *Quality and Reliability Engineering International*, vol. 16, 2000, pp. 325-335.