Discrete Event Systems Optimisation by inequalities

JUAN CARDILLO, FERENC SZIGETI
Control System Department
Los Andes University
Av. Tulio Febres Cordero, Facultad de Ingeniería, Escuela de Sistemas
Mérida – Venezuela

Abstract: The purpose of our paper is to develop a (symbolic) computational algorithm for the optimisation f a family of discrete events dynamic systems equipped with a multiobjective performance index. A discrete event version of the Pontryagin's Minimum Principle supports the algorithm. Unfortunately the miltiplier (dual variable) can depends on the variations of the optimal control, hence, the solution of the Pontryagin Minimum Principle requires treatments of a class a class varaitional inequalities, increasing the computational complexity. For this reason a special attention is paid to the reduction of the complexity.

Ke-Words: Optimisation. Discrete Event Dynamical Systems, Pontryagin Minimum Principle, Optimal Control, Controller Synthesis. Computational Complexity.

1 Introduction

Problems of optimisation, for discrete event process over discrete state space, arise very often in parallel computing, supervisor, and control, integration of industrial control systems, certain human systems, etc.

It is well know that the Pontryagin's Minimum Principle for discrete time process has certain geometric limitation. The Pontryagin's Minimum Principle is false, in general, without additional condition of convexity. Considering this fact, it is very surprising that a Pontryagin's Minimum Principle can be developed for a discrete event dynamical systems without any geometric structure. However, we have pay for this scheme in the greater complexity of the dual process, which will depend on both the optimal process and its variation. Hence instead of the pointwise optimisation of the Hamiltonian, a variational inequalities must be solve at each time. Variational inequalities are minimised by simbolic optimisation, which is a new with respect to the "classical " optimisation techniques of [2], [4], and [8].

Hybrid optimisation of plant performance, with respect to supervisory control can be reduced into discrete event optimisation.

Computer aided problem solving, tutoring, intelligent dialog can based in optimisation over the associated formal language as possible application, see [6].

2 A class of discrete event dynamical system

Let Ω be a finite alphabet of letters, events, etc. Ω^* denotes the set of all finite words, strings, including the empty string, θ. The concatenation of strings is an associative binary operation with the neutral element θ. Hence, Ω^* is a monoide. A subset $L \subset \Omega^*$ containing the empty string θ is a language.

2.1 Discrete dynamics induced by a language ($L \subset \Omega$)

Let $u \in L$. A string $v \in \Omega^*$ is active at u if $uv \in L$. Let us denote the subset of active strings at u, by L(u). Analogously, $w \in \Omega$ is an active event at $u \in L$, if the concatenation $uw \in L$. We notice that the letter w defines a word of one letter, hence uw is well defined. $\Omega(u) \subset \Omega$ is the subset of all active letters. The local dynamics is a partial mapping $f : Lx\Omega \to L$, $f(u,w) = uw$, for all pair (u,w) which satisfies the relation $w \in \Omega(u)$. Suppose that $u = u_0 u_1 \cdots u_{T-1} \in L$, $u_0, u_1, ..., u_{T-1} \in \Omega$. The words $x_i = u_0 u_1 ... u_{i-1}$, i=0,1,...T, are all prefixes of u. Suppose that $x_0 = \theta$, $x_1 ... x_T = u \in L$. Then u can be obtained by the difference equation $x_{i+1} = f(x_i, u_i), x_0 = \theta$.

Hence, we will suppose that all prefixes of all elements of the language L, also belong to L. In this case we say that L is prefix closed. If L is prefix closed then all words of L can be obtained by a difference equation in terms of the local dynamics f. In this case

$F(u^*, v) = f(f(...f(f(u^*, v_0), v_1),..., v_{T-2}), v_{T-1})$ (1),

defines a global dynamics, where $v = v_0 v_1 ... v_{T-1}$.

Consider a family $\Omega_0, \Omega_1, ..., \Omega_{T-1}$ of finite alphabets and the respective state spaces $X_0, X_1, ..., X_T$. A family of partially defined mappings gives the local dynamics in this case

$f_i : X_i x \Omega_i \to X_{i+1}$, i=0,1,...,T-1. (2)

$u \in \Omega_i$ is an active event, at the state $x \in X_i$, if the pair (x, u) belongs to the domain of f_i.

Let the subset of all active events be denoted by $\Omega_i(x)$, at all $x \in X_i$, i=0,1, ...,T-1.

Let $\xi \in X_i$ Then: $u = u_i u_{i+1} ... u_{T-1}, u_j \in \Omega_j$ (3), is an admissible string at ξ, if $u_i \in \Omega_i(\xi)$, $x_i = \xi$ and if for a $j \ge i$ x_j is defined by the difference equation

$x_{j+1} = f_j(x_j, u_j), x_i = \xi,$ (4)

then $u_j \in \Omega_j(x_j)$ and x_{j+1} can also be computed by the difference equation (4). The string $x_i x_{i+1} ... x_T$ of states is the trajectory, corresponding to the string $u = u_i u_{i+1} ... u_{T-1}$, which is called control. The pair (x, u) is a discrete process.

2.2 Discrete dynamical system induced by language.

Consider a language $L \subset \Omega^*$. Suppose that a function $\varphi : L \to Y$ is given. $u, v \in L$ are Nerode equivalent if L(u)=L(v), and $\varphi(uw) = \varphi(vw), w \in L(u) = L(v)$.

The classes of equivalence are the states. Over the obtained state space a (local) dynamics and an output mapping are induced. The obtained dynamical system is a realisation of the input-output mapping φ.

Now, we detail the construction of a graded realisation of the mapping φ. The length of a string is the number of events in it: $u = u_0 u_1 ... u_{T-1} \in L, u_i \in \Omega$, then the length of u, which is denoted by $|u|$, is T. $|\theta| = 0$.

$u, v \in L$ are equivalent if $|u| = |v|$, and u y v are Nerode equivalent. Let us denote the class of u by $[u]$, and the set of all classes of elements of length i, by X_i.

Of course, $x_0 = [\theta]$, consists of the unique string θ, and $X_0 = \{x_0\}$. The subset of the active strings at a state $x \in X_i, x = [u]$ can be defined by L(x)=L(u). Indeed, it does not depend on the representation $u \in x$.

Analogously, the set of active events at $x \in X_i$, is the subset $\Omega(x) = \Omega(u)$, if $x = [u]$.

The local dynamics $f_i : X_i x \Omega \to X_{i+1}$ is the partial mapping $f_i(x, u) = f_i([v], u) = [vu]$, defined over the subset of pairs (x,u), $u \in \Omega(x)$.

The length $|vu|$ =i+1, hence f_i maps into X_{i+1}. If Ω is a finite alphabet, then the graded dynamic system can be infinite, however all state spaces X_i are finite. The output mappings $\Phi_i : X_i \to Y$, are defined by $\Phi_i(x) = \Phi_i([u]) = \phi(u)$, which does not depend on the representation $x = [u]$.

3 Optimisation over discrete dynamical system

Consider a linear full ordering ≤ in Z^l. The ≤ is linear if the subset P{x≥0} of the nonnegative elements is a semigroup, with respect to the +, and the simplification property holds. Let us consider a graded discrete dynamical system $f_i : X_i x \Omega_i \to X_{i+1}$, with $\Phi_T : X_T \to Z^l$, multiobjective criteria.

Now, we define the problem of optimisation. We say that the admissible process (x^*, u^*), $u^* = u_0^* u_1^* ... u_{T-1}^*$, $x_{i+1}^* = f(x_i^*, u_i^*)$, $x_0^* = \xi \in X_i$, is optimal (minimal) if for all admissible processes (x,u), $j = |u|$, the inequalities $\Phi_{i+j}(x_j^*) \le \Phi_{i+j}(x_j)$ hold if $j \le T$ and $\Phi_{i+T}(x^*) \le \Phi_{i+T}(x_T)$ if $j > T$.

Of course, we can optimise among the processes of length T. Then, the optimality means that $\Phi_T(x_T^*) \le \Phi_T(x_T)$.

Now, suppose that we also have the function $g_i : X_i x \Omega_i \to Z^l$. Then the objective function can be defined by

$$J(x, u) = \Phi_T(x_T) + \sum_{i=0}^{T-1} g_i(x_i, u_i).$$

The admissible process (x^*, u^*) is optimal if for all admissible processes (x,u), $J(x^*, u^*) \le J(x, u)$.

It is well know that these two problems are equivalent. The virtually more general second problem can be transformed in to the first one.

3.1 Polynomial description of discrete dynamical systems

Our hypothesis is that the alphabets and the corresponding state spaces are finite. In this case all functions can be considered polynomials. For this, all sets of cardinality T can be identified by the set {0,1, ...,T-1}=Z_T.

Hence, if the cardinality of X_i, and Ω_i is n_i, and m_i respectively, then the dynamics is a partial mapping.

Using the Lagrange interpolation at all points of the domain of f_i, a polynomial representation for f_i is obtained. The polynomial is of degree at most $n_i - 1$ in the indeterminate x, and of degree at most $m_i - 1$ in the second indeterminate.

The formulation of the adjoint equation and the poof of the minimum principle involve the derivatives and gradients of the functions. However, we can not use these concepts. Replacing these classical tools

$u_0 = u^*_0, \cdots, u_{j-1} = u^*_{j-1}, u_j = v, u_{j+1} = u^*_{j+1}, ..., u_{T-1} = u^*_{T-1}$,

we introduce the concept of the Taylor residual. In the continuous case the Taylor residual of the smooth function f is defined by

$$f(y) - f(x) = Rf(x;y)(y-x) = \int f'(x+ty)\,dt\,(y-x).$$

For the monomial m(x,u)= $x^k u^l$, The Taylor residual is defined by

$$y^k v^l - x^k u^l =$$

$$\left(\sum_{i=0}^{k-1}\sum_{j=0}^{l}\frac{k}{i+j+1}\binom{k-1}{i}\binom{l}{j}(y-x)^i(v-u)^j x^{k-i-1}u^{l-j}\right)(y-x)$$

$$+\left(\sum_{i=0}^{k}\sum_{j=0}^{l-1}\frac{l}{i+j+1}\binom{k}{i}\binom{l-1}{j}(y-x)^i(v-u)^j x^{k-i}u^{l-j-1}\right)(v-u)$$

However the residuals are not unique, for example:

$$y^k v^l - x^k u^l = \left(\sum_{i=0}^{k-1} y^i x^{k-i-1} v^l\right)(y-x) + \left(\sum_{i=0}^{l-1} x^k u^i v^{l-i-1}\right)(v-u)$$

also holds. For our objective the equality $f(y) - f(x) = Rf(x;y)(y-x)$ has only importance, hence

$$Rm(x,u;y,v) = \left(\sum_{i=0}^{k-1} y^i x^{k-i-1} v^l, \sum_{i=0}^{l-1} x^k u^i v^{l-i-1}\right),$$

plays the role of the gradient of m.

For a vector function, the Taylor residual is analogous to the Jacobian. The partial residuals can also be defined, analogously, to the partial derivation.

3.2 The adjoint equation

Now, consider two process, (x^*, u^*), (x,u) of the some length T. Then, the adjoint equation is

$p_i = p_{i+1} R_x f_i(x^*_i, u_i; y_i)$, $p_T = R\Phi_T(x^*_T, x_T)$,

where the trajectory $p_0 p_1 ... p_T$ of the adjoint equation belong to $R^{lxn_{j+i}}$.

In terms of the adjoint trajectory, the value of the residual of the multiobjective function at the final time is given for

$$\Phi_T(x_T) - \Phi_T(x^*_T) = \sum_{j=0}^{T-1} p_{j+1} f_j(x^*_j, v_j) - \sum_{j=0}^{T-1} p_{j+1} f_j(x^*_j, u^*_j).$$

Now, fix an j and consider two admissible processes (x^*, u^*) and (x,v), such that $u^*_0 = v_0$, $u^*_1 = v_1$,..., $u^*_{j-1} = v_{j-1}$, $u^*_j = v$, $u^*_{j+1} = v_{j+1}$,..., $u^*_{T-1} = v_{T-1}$. Then, the residual of the multiobjective function at the final time is

$$\Phi_T(x_T) - \Phi_T(x^*_T) = p_{j+1} f_j(x^*_j, v_j) - p_{j+1} f_j(x^*_j, u^*_j).$$

3.3 The Extension of Pontryagin's Minimum Principle.

Define the Hamiltonian function by $H(j, x^*, u^*, p) = pf_j(x^*, u^*)$.

Suppose that x^*, u^* is an optimal process among the admissible processes of length T. If $u^*_0 = u_0$, $u^*_1 = u_1$,..., $u^*_{j-1} = u_{j-1}, u^*_j = v$, $u^*_{j+1} = u_{j+1}$,..., $u^*_{T-1} = u_{T-1}$, $x^{j,v}$ is the corresponding trajectory, $p^{j,v} = p^{j,v}_0 ... p^{j,v}_T$, is the solution of the adjoint equation corresponding to the pair of processes (x^*, u^*), (x,u). Then

$$H(j, x^*_j, u^*_j, p^{j,v}_{j+1}) \le H(j, x^*_j, v, p^{j,v}_{j+1})$$

We notice that the difference equation and the adjoint equation can be rewritten in terms of the Hamiltonian:

$$x^*_{j+1} = R_p H(j, x^*_j, u^*_j, p^{i,v}_{j+1}),$$

$$p^{i,v}_j = R_x H(j, x^*_j, u^*_j, p^{i,v}_{j+1}, x^{i,v}).$$

4 Symbolic algorithm for solution PMPE

Now, we show a symbolic algorithm in order to solve the integer Pontryagin's Minimum Principle. At each step we need two routine.

1. The first reduces the degree of the polynomial, via the Langrage interpolation.

2. The second routine is the kernel of the algorithm. It consists in the optimisation of variational inequality, of the form:

$$0 \leq p(\xi, u^*, u)\left[f(\xi, u) - f(\xi, u^*)\right],$$

where $x \in X$ is a given state. We say that $u^* \in \Omega(\xi)$ is a solution of the inequality, if it holds for all $u \in \Omega(\xi)$. In general, suppose that $\Psi(\xi)$ is a solution of the variational inequality; then we can define the mapping $\Psi : X \to U$. Of course, Ψ can be multifunction.

4.1 Symbolic algorithm

Our purpose is to solve the Pontryagin's Minimum Principle in terms of the solutions Ψ_i of the variational inequalities. Then, we also show an algorithm to compute Ψ_i

Steps 1: i) Let us compute $x^*(T) = f(x^*(T-1), u^*(T-1))$. ii) u is replaced by v in the previous expression for t=T-1, that is, $x^{T-1,v}(T) = f(x^*(T-1), v)$.

iii) Compute the adjoint equation:

$$p^{T-1,v}(T) = R\Phi(x^*(T), x^{T-1,v}(T)) =$$
$$= R\Phi(f(x^*(T-1), u^*(T-1)), f(x^*(T-1), v)).$$

iv) The Hamiltonian

$$0 \leq p^{T-1,v}(T)[f(x^*(T-1), v) - f(x^*(T-1), u^*(T-1))] =$$
$$= R\Phi[f(x^*(T-1), u^*(T-1)), f(x^*(T-1), v)]*$$
$$\left[f(x^*(T-1), v) - f(x^*(T-1), u^*(T-1))\right]$$

v) Let us apply Routine 1 to reduce degree of the polynomial

vi) Applying Routine 2, $\Psi = \Psi(T-1, \xi)$, is obtained such that $u^*(T-1) \in \Psi(T-1, x^*(T-1))$ satisfies the above inequality.

$\vdots$

Steps T-i: Suppose that for i, $\Psi(T-1,\xi)...\Psi(i+1,\xi)$ are constructed. Then, by induction it can be proven that: (1) $x^*(i+1), x^*(i+2), ..., x^*(T)$,

(2) $x^{i,v}(i+1), x^{i,v}(i+2), ..., x^{i,v}(T)$,

(1) are polynomials of $x^*(i)$ and $u^*(i)$ and (2) are polynomials of $x^*(i)$ and $u^*(i)$ and v. Therefore $p^{i,v}(i+1), p^{i,v}(i+2), ..., p^{i,v}(T)$ are also polynomials of $x^*(i)$ and $u^*(i)$ and v. Apply routine 1, in order to reduce the degree of the polynomials. Let us apply routine 2, a (possible multiobjective) function $\Psi = \Psi(i, \xi)$ is obtained, such that $u^*(i) \in \Psi(i, x^*(i))$ satisfies the above inequality.

Hence

$$0 \leq p^{i,v}(i+1)\left[f(x^*(i), v) - f(x^*(i), u^*(i))\right] =$$
$$p(x^*(i), u^*(i), v)\left[f(x^*(i), v) - f(x^*(i), u^*(i))\right]$$

Applying routine 1, a polynomial function $\Psi = \Psi(i, \xi)$ is obtained, such that $u^*(i) \in \Psi(i, x^*(i))$.

The construction of the optimal processes in term of the functions $\Psi(0,\xi), \Psi(1,\xi,), ..., \Psi(T-1,\xi)$ is easy:

$$x^*(0) = \xi,\ u^*(0) = \Psi(0, \xi),$$
$$x^*(1) = f(x^*(0), u^*(0)),\ u^*(1) = \Psi(1, x^*(1)),$$
$$\vdots$$
$$x^*(i+1) = f(x^*(i), u^*(i)),\ u^*(i+1) = \Psi(1, x^*(i+1)),$$
$$\vdots$$

Therefore, the optimal control and optimal trajectory are obtained, for each instant of time. The construction shows above can be considered as the generalisation of the Riccati equation for discrete process.

5 Optimisation based on the solution of Variational Inequalities

In the previous session a symbolic algorithm was shown to obtain optimal controls. The symbolic algorithm was based in two routines. The first routine reduces the degree of the polynomials, obtained from the Hamiltonian function (PMPE). The second routine is an algorithm that gives the solution to the Pontryagin Minimum Principle in terms of the function Ψ. In [1], we have designed a tool in Mathematica in where the VI is calculated for each instant of time. In this session is given an algorithm in order to compute the solution to the set of VI.

5.1 The min-max problem

The set of VI, in principle, can be converted in a min-max problem. Then, optimisation is achieved by a two phases method. In the first phase a minimum is calculated. In the second phase a maximum is calculated and if the maximum is less than zero, the VI has solution(s), that is, our algorithm result will also give up a criterion for the existence of the solution of the min-max problem.

The principal idea of the optimisation will be presented for a Boolean function f, defined over the set {-1,1}. Then, defining a modified sign function by

$$\mathrm{sign}(t)=\begin{cases}-1 \text{ if } t\le 0\\ 1 \text{ if } t\ge 0,\end{cases}$$

$u^*\in \mathrm{sign}(f(-1)-f(1))$ are the points, where f reaches its minimum. We remark, that sign(0)={-1,1}.

Now, let us suppose that the cardinality of the domain D of f is 2^{k+1}. Then, the bijection between {0,1,..., 2^{k+1}-1} and the set of the k+1-vectors $(u_0,u_1,...,u_k)\in\{-1,1\}$, is defined by

$$1/2\big((1-u_0)+2(1-u_1)+...+2^k(1-u_k)\big).$$

Hence, the function f can be considered as a function of k+1 variable over $\{-1,1\}^{k+1}$:

$$(u_0,u_1,...,u_k)\to f(u_0,u_1,...,u_k)=f_0(u_0,u_1,...,u_k)$$

1.a) Fixed $u_0,...,u_{k-1}$, the minimum of the function $u_k\to f(u_0,...,u_{k-1},u_k)$ can be expressed by the multifuction:

$u_k^*(u_0,u_1,...,u_{k-1})=\mathrm{sign}[f_0(u_0,u_1,...,u_{k-1},-1) - f_0(u_0,u_1,...,u_{k-1},1)]$.

In fact, if

$f_0(u_0,u_1,...,u_{k-1},-1) = f_0(u_0,u_1,...,u_{k-1},1)$, then $u_k^*(u_0,u_1,...,u_{k-1})=\{-1,1\}$ for all values of $u_0,u_1,...,u_{k-1}$ (f_0=constant).

1.b) Let us define

$f_1(u_0,u_1,...,u_{k-1})=f_0(u_0,u_1,...,u_{k-1}, u_k^*(u_0,u_1,...,u_{k-1}))$.

2) The general construction. Let us suppose, that the ith function $f_i:\{-1,1\}^{k+i-1}\to\Re$ is defined. Then, for fixed $(u_0,u_1,...,u_{k-1})\in\{-1,1\}^{k-i}$, let us consider the function $u_{k-i}\to f_i(u_0,...,u_{k-i-1},u_{k-i})$. That achieves its minimum at $u_{k-i}^*(u_0,...,u_{k-i-1})=\mathrm{sign}[f_i(u_0,...,u_{k-i-1},-1)-f_i(u_0,...,u_{k-i-1},1)]$.

Then, we can define the (multi-)function $f_{i+1}(u_0,...,u_{k-i-1})=f_{i+1}(u_0,...,u_{k-i-1}, u_{ki}^*(u_0,...,u_{k-i-1}))$.

The algorithm is defined if:

1. i=k, and $u_0\to f_k(u_0)$ achieves its minimum at $u_0^*\in \mathrm{sign}[f_k(-1)-f_k(1)]$.

2. i<k, $f_i(u_0,u_1,...,u_{k-i-1},-1) = f_0(u_0,u_1,..., u_{k-i-1},1)$. Then, for all $(u_0,u_1,...,u_k)\in\{-1,1\}^{k-i+1}$ f_i is constant and at all points is minimal.

$u_0^*\in \mathrm{sign}[f_k(-1)-f_k(1)]\subset\{-1,1\}$

$$(u_0^*,u_1^*)=\{(u_0,u_1);u_0\in u_0^*,$$
$$u_1^*\in \mathrm{sign}(f_{k-1}(u_0,-1)-f_{k-1}(u_0,1))\}\subset\{-1,1\}^2$$

$\vdots$

$$(u_0^*,u_1^*,\cdots,u_{k-i}^*)=\{(u_0,u_1,\cdots,u_{k-i});$$
$$(u_0,u_1,\cdots,u_{k-i-1})\in(u_0^*,u_1^*,\cdots,u_{k-i-1}^*),$$
$$u_{k-i}^*\in \mathrm{sign}(f_i(u_0,u_1,\cdots,u_{k-i-1},-1)-f_i(u_0,u_1,\cdots,u_{k-i-1},1))\}$$
$$\subset\{-1,1\}^{k-i}$$

The set of all minimal solution is $(u_0^*,u_1^*,\cdots,u_{k-i}^*)\subset\{-1,1\}^{k+1}$ defined recursively.

At the first step the VI which define Ψ_{T-1}, has the form p(v, v) ≥ p(u*, v) (1).

The other VI has the form

p(u*,v, v) ≥ p(u*, u*, v) (2).

Both inequalities can be solved by using the given optimisation procedure:

(i) Let us fix u*. Then, consider the function $v\to p(v,v)-p(u^*,v)$, and compute its minimum as the function of u*:

$$\min_v(p(v,v)-p(u^*,v))=P(u^*)$$

u^*

If there exists u* such that $0\le P(u^*)$, then u* is the solution of the VI. However, all solutions of the VI are those u*, which satisfy the inequality $0\le P(u^*)$.

(ii) The solution of (2) can be obtained analogously. Let us fix u* and consider the function $v\to p(u^*,v,v)-p(u^*,u^*,v)$ then

$\min_v(p(u^*,v,v)-p(u^*,u^*,v))=P(u^*)$, u* is solution of (2) if only if $0\le P(u^*)$.

We notice that, if $\max_{u^*}P(u^*)<0$, then the VI have no solutions.

2. The direct implementation of that algorithm can be result overflow of the memory for relative small problem.

6 Reduction of the complexity

In this section a computational implementation will show, after a carefull analysis of the symbolic algorithm. The computation will be realised in two states: (i) reduction of the complexity by manipulation of the polynomials in question, (ii) using explicit symbolic expressions.

6.1 Reduction of the complexity by manipulation of polynomials.

Let us consider a generic monomial $u_0^{\alpha_0}u_1^{\alpha_1}\cdots u_k^{\alpha_k}$ of the polynomial $p(u_0,\cdots,u_k)$. The domain of the polynomial function defined by the polynomial $p(u_0,\cdots,u_k)$ is $\{-1,1\}^{k+1}$, therefore, for all vectors $u\in\{-1,1\}^{k+1}$, $u_0^{\alpha_0}u_1^{\alpha_1}\cdots u_k^{\alpha_k} = u_0^{\overline{\alpha}_0}u_1^{\overline{\alpha}_1}\cdots u_k^{\overline{\alpha}_k}$, where

$$\overline{\alpha}_i = \begin{cases} 0, \text{if } \alpha_i \text{ is even} \\ 1, \text{if } \alpha_i \text{ is odd} \end{cases}.$$

Hence $p(u_0,\cdots,u_k) = \sum_{\{i_1\cdots i_l\}\subset\{0,\cdots,k\}} a_{i_1\cdots i_l} u_{i_1}\cdots u_{i_l}$ (3).

For example

$$(2u_1+u_2)^{13} = \sum_{i=0}^{13}\binom{13}{i}2^i u_1{}^i u_2{}^{13-i} =$$

$$= \left(\sum_{i=0}^{6}\binom{13}{2i}2^{2i}\right)u_2 + \left(\sum_{i=0}^{6}\binom{13}{2i+1}2^{2i+1}\right)u_1$$

The general form of the polynomials of 2 indeterminates, after the described reduction can be written in

$$g(u_0,u_1) = a_{11}u_0u_1 + a_{10}u_0 + a_{01}u_1 + a_{00}.$$

Let us consider the subset $\underline{i} = \{i_1,\cdots,i_l\}\subset\{0,1,\cdots,k\}$, which is naturally ordered $i_1 < i_2 < \cdots < i_l$. Then, there exits a natural bijection of the set of all subsets $\underline{i}\subset\{0,1,\cdots,k\}$ into the set of all binary numbers of k+1 digits, by

$$\underline{i} \leftrightarrow 0\cdots010\cdots010\cdots010\cdots0 = B(\underline{i})$$

1 $\quad i_1+1 \quad i_2+1 \quad i_l+1 \quad k+1$

Let $u_{\underline{i}} = \prod_{j=1}^{l} u_{i_j}$. Then, the polynomial $p(u_0,\cdots,u_k)$ can be rewritten by

$$p(u_0,\cdots,u_k) = \sum_{\underline{i}\subset\{0,\cdots,k\}} a_{B(\underline{i})}u_{\underline{i}} = \sum_{b=0\cdots0}^{1\cdots1} a_b u_{B^{-1}(b)}.$$

6.2 The algorithm

Let $g_{n+1}(u_0,u_1,...,u_n)$ be a polynomial. Then,

$$u_n^* = -\mathrm{Sign}\left[g_{n+1}(u_0,...,u_{n-1},1) - g_{n+1}(u_0,...,u_{n-1},0)\right].$$

For $i\in\{0,1,...,k\}$,

we will define new polynomial, $g_{k-i+1}(u_0,u_2,...,u_{k-i})$, with 2^{k-i} coefficient from the lineal relationships following:

$$g_{k-i}(u_0,u_1,...,u_{k-i}) = g_{k-i+1}(u_0,u_1,...,u_{k-i}, u_{k-i+1}^*(u_0,u_1,...,u_{k-i}))$$

Evaluate in domain, we have:

$$g_{k-i}(-1,-1,...,-1) = g_{k-i+1}(-1,...,-1,u_{k-i}^*(-1,...,-1))$$

$$\vdots$$

$$g_{k-i}(1,1,...,1) = g_{k-i+1}(1,...,1,u_{k-i}^*(1,...,1))$$

Rewriting, we have

$$\begin{pmatrix} -A_{k-i-1} & A_{k-i-1} \\ A_{k-i-1} & A_{k-i-1} \end{pmatrix}\begin{pmatrix} a_{b1...1} \\ \vdots \\ a_{b0...0} \end{pmatrix} = \begin{pmatrix} g_{k-i+1}(-1,...,-1,u_{k-i+1}^*(-1,...,-1)) \\ \vdots \\ g_{k-i+1}(1,...,1,u_{k-i+1}^*(1,...,1)) \end{pmatrix}$$

where $A_1 = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$.

Then

$$u_{k-i}^* = -\mathrm{Sign}\left[g_{k-i}(u_0,u_1,...,u_{k-i-1},1) - g_{k-i}(u_1,u_2,...,u_{k-1-1},0)\right]$$

7 Example

Let us f(u) a polynomial function over the domain $D=\{0,1,2,3,4,5,6,7\}$. The transformed domain D could be represented by $u_1,u_2,u_3\in U=\{-1,1\}$ such that:

$0\approx\{-1,-1,-1\}, 1\approx\{-1,-1,1\}, 2\approx\{-1,1,-1\}, 3\approx\{-1,1,1\}$,

$4\approx\{1,-1,-1\}, 5\approx\{1,-1,1\}, 6\approx\{1,1,-1\}, 7\approx\{1,1,1\}$,

The matrix of the linear system equation is:

$$\begin{pmatrix} -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{111} \\ a_{110} \\ a_{101} \\ a_{100} \\ a_{011} \\ a_{010} \\ a_{001} \\ a_{000} \end{pmatrix} = \begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{pmatrix} = \begin{pmatrix} g(-1,-1,-1) \\ g(-1,-1,1) \\ g(-1,1,-1) \\ g(-1,1,1) \\ g(1,-1,-1) \\ g(1,-1,1) \\ g(1,1,-1) \\ g(1,1,1) \end{pmatrix}$$

$\begin{pmatrix} -A_2 & A_2 \\ A_2 & A_2 \end{pmatrix} = A_3$. Then, the polynomial g(u) is given for:

$$g(u_1, u_2, u_3) = a_{111}u_1u_2u_3 + a_{110}u_1u_2 + a_{101}u_1u_3 + a_{100}u_1 + a_{011}u_2u_3 + a_{010}u_2 + a_{001}u_3 + a_{000}$$

The running of algorithm in Mathematica produce the following result.

Polynomial :a[0] + a[4] u[1] + a[2] u[2] +
a[6] u[1] u[2] + a[1] u[3] + a[5] u[1] u[3] + a[3] u[2] u[3] + a[7] u[1] u[2] u[3]

****

u*[3] :-Sign[a[1] + a[5] u[1] + a[3] u[2] +
a[7] u[1] u[2]]

Auxiliary polynomial
b[0]+b[2] u[1] + b[1] u[2] + b[3] u[1] u[2]

****

u*[2] :-Sign[a[2] + a[6] u[1] + Sign[a[1] + a[5] u[1]] (a[1] + a[5] u[1]) - Sign[a[1] + a[3] + a[5] u[1] + a[7] u[1]] (a[1] + a[3] + a[5] u[1] + a[7] u[1])]

Auxiliary Polynomial : b[0] + b[1] u[1]

****

u*[1] :-Sign[a[4] + a[1] Sign[a[1] - a[3] Sign[a[2] + a[1] Sign[a[1]] - (a[1] + a[3]) Sign[a[1] + a[3]]]] + Sign[a[2] + a[1] Sign[a[1]] - (a[1] + a[3]) Sign[a[1] + a[3]]]
(a[2] - a[3] Sign[a[1] - a[3] Sign[a[2] + a[1] Sign[a[1]] - (a[1] + a[3]) Sign[a[1] + a[3]]]]) - a[1] Sign[a[1] + a[5] - a[3] Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]] - a[7] Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]]] - a[5] Sign[a[1] + a[5] - a[3] Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]] - a[7] Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]]] - Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]] (a[2] + a[6] - a[3] Sign[a[1] + a[5] - a[3] Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]] - a[7] Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]]] - a[7] Sign[a[1] + a[5] - a[3] Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]] - a[7] Sign[a[2] + a[6] + (a[1] + a[5]) Sign[a[1] + a[5]] - (a[1] + a[3] + a[5] + a[7]) Sign[a[1] + a[3] + a[5] + a[7]]]])].

8 Conclusion

In this paper, The method is an algorithm is presented for the optimisation of a class of DEDS equipped with multiobjective criteria based on the solution of VI obtained from the Pontryagin's Minimum Principle.
These algorithms have explicit symbolic expression. Hence we can calculate the optimal solutions by substitution.

Reference:

[1] J. Cardillo " Optimización de sistemas a eventos discretos usando el Principio de Mínimo de Pontryagin", Universidad de los Andes, Facultad de Ingeniería, Postgrado Ingeniería de Control, Mérida, Venezuela, Marzo 1994.

[2] K.V.Mital " Métodos de Optimización en Investigación de Operaciones y Análisis de Sistemas, Universidad de Roorkee, Editorial Limosa, México, 1984. Pp 211-237.

[3] Caines, P. E. y Wei Y. J., On MarKovian Fragments of Cocolog for logic control systems, Proceedings of the 31st IEEE CDC, Tucson, AZ, December 1.992.

[4] D. Fayard and G. Platean, Resolution of 0-1 problem Knapsack Problem Comparison of Methodos, Mathematical Programming, vol. 8, 1975.

[5] O. V. German y D. V. Ofitserov, Problem Solving: Methods Programing and future Concepts, Washintong: State University of Informatics and Radio Electronics Minsk, Republic of Belarus Catholic University of America, 1995.
[6] P. J. Ramadge, and W. M. Wonham, Supervisor control of a class of discrete event process, Siam J. Control and Optimisation, vol. 25, Nº 1, January 1987.
[7] W. M. Wonham and P. J. Ramadge, On the supremal controllable sublanguage of a given language, Siam J. Control and Optimisation, vol. 25, Nº 3, May 1987.
[8] Raja Sengupta, and Stephanie Lafortune, An optimal control theory for dicrete event systems, Siam J. Control and Optimisation, vol. 36, Nº 2, January 1998.