The context.
I solve numerically an ODE system which depends on **56** parameters.
What I am interested in is essentially the times at which different events appear.
Most specifically I want to know the maximum occurence time for a particular event as the parameters vary (independently the one of the other) in an hyperbox.

How did I attack the problem?
Let **sys** the ODE system to solve, **evs** a list of events, **pars** a collection of parameters **sys** depends upon, and **{x(t), v(t)}** the two unknown functions solutions of **sys.**
Let **opt** a list of **dsolve** options.
The solution sol is the procedure returnd by

sol := dsolve(sys, numeric, parameters=pars, events=evs, options=opt)

I'm focusing on a particular event **MyEvent** of the form **[x(t)-u, halt]**.
Here **u** is a quantity which depends upon some parameters in pars.

For the specific case I'm concerned with, it has been decided that **31** parameters ou**t** of **56** will be set to constant values and that the **25** remaining ones will take values in given ranges.
To clarify this, let q[n] any of these **25** variable parameters: it is assumed that **q[n]** can take equally likely any value in a range **LB[n]..UB[n]** (**LB** and **UB** means Lower Bound and Upper Bound respectively).
These **25** parameters are mutually independent in the sense that the value that any one of them takes does not depend on the values taken by the others.

In the list of events, **MyEvent** has number **9** (thus the ad hoc coding below)
I have written a procedure **OBJ** which takes as input a list (or vector) **q** of length **25** and returns the time at which event **9** is triggered.

```
OBJ := proc(q)
  local tps, k, jeu, tps_decol:
  jeu := copy(ref):
  for k from 1 to nva_va do
    jeu[var[k]] := q[k]
  end do:
  jeu       := convert(jeu, list);
  tps_decol := eval(tto, va =~ jeu);
  jeu       := [ 0, jeu[], tps_decol, 1]:
  sol(parameters=jeu):
  sol(tps_max);
  sol(eventfired=[9])[1]
end proc:
```

In this procedure **ref** contains "reference" values for all the **56** parameters.
The loop over **k** just assign new values to the **nva_va=25** variable parameters whose numbers are defined by **var**.
tps_decol correspond to the time at which the initial conditions
(**x(tps_decol)=v(tps_decol)=0**) are imposed.

Preliminary results
Let **T** the time at which event **9** occurs.
From physical considerations on may infer that **T** is an increasing function of some of the **25** variable parameters, a decreasing function of some others and that it exists a few parameters we don't know the way they act.
This expert analysis leaves us with **9** parameters for which we do not know if their increase increase **T** or decrease it.
To clear up the problem we have fixed all the **25-9=16** parameters to their values (**LB** or **UB**) infered from expertise, and we have defined a 29 full factorial design https://www.itl.nist.gov/div898/handbook/pri/section3/pri3332.htm to investigate the role of the **9** "we-do-not-know-how-they-act" parameters.

From this design (whose justification relies upon some hypothesis I'm not going to discuss here) it appears that a situation, which is a corner of the **25** dimensional hyperbox, has given a value **T = 100.529 s**.
I coded a simple Newton-like method to go further and seek, starting from this corner, an even larger value of **T**.
I got **T =100.635 s**.
The corresponding **25**-dimensional point is named **best** in the attached files.

What I would like is to verify if that this value of T is indeed the highest value that can be reach when the **25** variable parameters vary within the hyperbox.
I thought using **Optimization:-NLPSolve** but It keeps returning me a "no improved point could be found" error despite all my attempts.