PARETO BI-CRITERION OPTIMIZATION FOR SYSTEM SIZING : A DETERMINISTIC AND CONSTRAINT BASED APPROACH

Pierre-Alain Yvars[1]
(1)LISMMA, SupMeca, France.

ABSTRACT
In this paper we are studying a deterministic constraint based approach to solve Pareto bi-criterion optimization problems in design. After presenting the use of multi-objective optimization methods in design, the CSP method and the several ways to solve it is introduced. A quick overview of CSP application in product engineering is given too. Moreover, we introduce an optimization point of view for CSP and we propose a deterministic alternative to stochastic methods for solving Pareto bi-objective system sizing problems. An example in mechanical system optimization is given via the case study of the Pareto bi-criterion optimal design of a bolt coupling. The case is modeled as a Constraint Satisfaction Problem on both discrete and real variables. Finally, the numerical results and the Pareto frontier are exposed.

Keywords: Constraint propagation, CSP, Pareto frontier, bi-criteria optimization, system sizing.

1 INTRODUCTION

The context of integrated and collaborative design of mechanical product as became now the usual context of design. Even if some concepts and tools are available to support this process, lacks still exist. In this context optimization of product remains a strong issue. In this paper we are studying a deterministic constraint based approach to solve Pareto bi-criterion optimization problems in design.
A first naïve approach should be used to deal with CSP and multi-objective optimization. It consists on transforming the interval of values of the first criteria into a set of discrete values. For each of them, a mono-objective algorithm is started to minimize the value of the second criteria. Unfortunately, this approach has at least two main drawbacks in design: On the one hand, there are many problems to adjust the set of discrete values and on the other hand, in case of a non convex and discontinuous design problem, we can miss several optimal points. The main purpose of this article is to go over these limitations in case of mixed design problems.
After presenting the use of multi-objective optimization methods in design, the CSP method and the several ways to solve it is introduced. A quick overview of CSP application in product engineering is given too. Moreover, we introduce an optimization point of view for CSP and we propose a deterministic alternative to stochastic methods for solving Pareto bi-objective system sizing problems. An example in mechanical system optimization is given via the case study of the Pareto bi-criterion optimal design of a bolt coupling. The case is modeled as a Constraint Satisfaction Problem on both discrete and real variables. Finally, the numerical results and the Pareto frontier are exposed.

2 MULTI-OBJECTIVE OPTIMIZATION IN DESIGN

2.1 Multi-objective optimization problem

A lot of problems in the design fields are relevant to multi-objective optimization problems. Multi-objective optimization is an answer to the need of satisfying both many conflicting criterion. Because there is rarely a solution better than another at any point, different compromises depending on preferences can be chosen.

A multi-objective problem is defined such that:

$X = (x_1, x_2, \dots, x_n)$ called decision vector. In a design problem, the x_i variables are called the design parameters.

$F = (f_1, f_2, \dots, f_m)$ called performance vector. Of course, in mono-objective optimization problem, F is a scalar value.

$$\exists p \in \mathbb{N}, \forall j \in \{1, \dots, p\}, \exists g_j : \mathbb{R}^n \to \mathbb{R}, \exists \mathcal{X} \subseteq X, g_j(\mathcal{X}) = 0 \quad (3)$$

$$\exists q \in \mathbb{N}, \forall j \in \{1, \dots, q\}, \exists h_j : \mathbb{R}^n \to \mathbb{R}, \exists \mathcal{X} \subseteq X, h_j(\mathcal{X}) \leq 0 \quad (4)$$

$$\forall i \in \{1, \dots, m\} f_i : \mathbb{R}^n \longrightarrow \mathbb{R}^+ \quad (5)$$

$$Find \quad X / \min F \quad (6)$$

Unfortunately, the solutions to this problem rarely minimizes all the f_i. It is necessary to propose a comparison operator to determine if a performance vector is better than another or if they are equivalents. A possibility is to use the relation of domination according to the definition given by Pareto. Noting $\preccurlyeq$ this relation in its wide sense and $\prec$ in its strict sense, F_i dominates F_j in Pareto sense if and only if:

$$\forall k \in \{1, \dots, m\}, \ f_{ik} \preccurlyeq f_{jk} \quad (7)$$

and

$$\exists k \in \{1, \dots, m\}, \ f_{ik} \prec f_{jk} \quad (8)$$

2.2 Application in design

The problem consists on determining the non-dominated set of points in the performance space. For any m, the Pareto hyper-surface can theoretically be obtained although its calculation is usually difficult and expensive.

More often, design problems and more precisely sizing ones are characterized by:

- A strong set of requirements.
- Mixed variables: the design parameters and the performance parameters should be both continuous and discrete.
- Many analytical relations: The parameters have to satisfied many algebraic relations.

A lot of works have been done to use stochastic algorithms to solve multi-objective optimization problems in design. A genetic algorithm as NSGA II give the opportunity to draw a good approximation for the Pareto frontier [1]. Unfortunately, those algorithms have several drawbacks: the calculation time is very high and more often they check the satisfaction of the analytical relations of the design problem by using a penalty mechanism.

The constraint based approach, that we will deal with in this article, allows us to express and check the analytical relations posted on the design parameters and the performance parameters. This analytical relations should be all algebraic equations and inequalities.

3 BI-CRITERION OPTIMAL SIZING AS A CSP PROBLEM

3.1 CSP

A CSP is defined by a Triplet (X, D, C) such that [2]:

- $X = \{X_1, X_2, \dots, Xn\}$ is a finite set of variables called constraint variables with n being the integer number of variables in the problem to be solved.

- $D = \{D_1, D_2, \dots, D_n\}$ is a finite set of variables value domains of X such that :

$$\forall i \in \{1,..,n\}, x_i \in Di \quad (9)$$

A domain should be a real interval or a set of discrete values.

- $C = \{C_1, C_2, \dots, Cp\}$ is a finite set of constraints, p being any integer number representing the number of constraints of the problem.

$$\forall i \in \{1,..,n\}, \exists \mathcal{X}_i \subseteq X \ / \ C_i(\mathcal{X}_i) \quad x_i \in D_i \quad (10)$$

A constraint is any type of mathematical relation (linear, quadratic, non-linear, Boolean…) covering the values of a set of variables.

More precisely, the constraints can be the following:

- Logical: such that x=1 or y=4; x=3 ⇒ z=5
- Arithmetical: such that x > y; 2x+3y < z

- Non-linear: such that cos(x) < sin(y)
- Explicit: in the form of n-tuples of possible values such that: (*x*, *y*) (0, 0), (1, 0), (2, 2)
- Complex: such that: the variable values x, y, z must all be different.

The variable domains can be:
- Discrete: in the form of sets of possible values.
- Continuous: in the form of intervals on real numbers

Solving a CSP boils down to instantiating each of the variables of *X* while meeting the set of problem constraints *C*, and at the same time satisfying the set of problem constraints *C*.

The solving process for a CSP depends on the type of the constraint variables. In fact CSP on integer variables called discrete CSP are different from CSP on real variables also called continuous or numerical CSP.
- On the one hand, for solving discrete CSP, the methods are ones arising from operational research and artificial intelligence. The first work dates back over thirty six years [3]. These discrete CSP methods, of exponential complexity, are based on enumeration and filtering. This filtering, also called constraint propagation, enables the definition domains of variables to be reduced as the resolution process evolves.
- On the other hand, CSPs have been developed with real variables with values in intervals. This interval-based resolution technique is a synthesis between interval-based analysis [4] and CSPs in [5,6]. Several techniques have been developed, one of which is presented as an example in [7].

During the design process, designers used and managed design rules, tables, abacus, relations…All these structures should be modeled as constraints (mathematical relations between variables).
The CSP community has developed work applicable in product and systems design [8-11].

3.2 Numerical CSP solving process

Search algorithms as Branch and Prune start the process by selecting a variable to bisect. The order in which this choice is done is referred the variable ordering. A correct ordering decision can be crucial to perform an effective solving process in case of real-life problems. There exist several heuristics for selecting the variable ordering. After selecting the variable to bisect, the algorithms have to select a subinterval form the variable's domain. This selection is called the value ordering. It can also have an important impact on the duration of the solving process. The Prune subroutine contracts the intervals of D by using interval arithmetic [4] and consistency mechanisms [6]. The goal is to fit the intervals bounds as much as possible without losing any solutions.

```
BP(CSP(X,D,C),{})
begin
D ← Prune(C,D)
if notEmpty(D) then
 if OkPrecise(D) then Insert(D,L)
   else
   (D1,D2) ← Split(D,ChooseVariable(X))
   BP(CSP(X,D1,C),L)
   BP(CSP(X,D2,C),L)
 endif
return L
end
```

Figure 1. Branch and Prune algorithm[7]

3.3 A deterministic constraint based Pareto bi-criterion optimization algorithm

The mono-objective optimization principle adopted to minimize the value of an objective f is described on Figure 2. Usually, f should be a variable equal to a constraint expression representing the criteria to minimize. The key point is to solve by dichotomy a sequence of CSP where the set of constraints increases from one CSP to the next. At each step, we add a constraint expressing that the next CSP has to be better than the current one according to the minimization of the f variable. The process stops on the CSP which minimize the f variable value when the required precision ε is reached. This is a kind of branch and bound method which works on finite domains as well as intervals.

OptimCSP(*X* , *D*, *C*)
begin
$f \in [f_{min}, f_{max}]$
CSP ← *(X, D, C)*
while $f_{max} - f_{min} > \varepsilon$
$C \leftarrow C \cup \{f < \frac{(f_{max} + f_{min})}{2}\}$
if *find a solution for CSP*
$f_{max} \leftarrow fval$
else
$C \leftarrow C - \{f < \frac{(f_{max} + f_{min})}{2}\}$
$f_{min} \leftarrow \frac{(f_{max} + f_{min})}{2}$
endIf
endWhile
return $[f_{min}, f_{max}]$
end

Figure 2: CSP and mono criteria optimization

Usually, the problem of finding the non-dominated frontier is addressed by finding the optimal value in one direction and restart the search with constraints that limit the search space to other Pareto-Optimal solutions. For a bi-criterion problem, a naïve approach consists on solving a mono-objective problem for each value of the second criteria but a lot of drawbacks should appear especially when the Pareto frontier is not convex and/or discontinuous.

Van Wassenhove and Gelders [12] find the non dominated frontier in a bi-criterion (f_1, f_2) scheduling problem. It should be described as follow:

1st step: find the optimal solution (Opt_2) that minimize function f_2; Let d 🡐 Op_{t2}
2nd step: impose that $f_1 < f_1(d)$
3rd step: minimize f_2. If a solution S exists, then it is non dominated; else exit.
4th step: Let d 🡐 $f_1(S)$. Go to 2nd step.

If we try to use both csp and Van Wassenhove & Gelders principles we obtain what we call the BiPareto algorithm as shown in Figure 3.

```
BiPareto(X, D, C, f1, f2)
begin
Opti(X, D, C, f2)
repeat
      Nondominated ← NonDominated ∪ (X,D,C)
      C ← C ∪ (f1 < f1val)
until ! Opti(X, D, C, f2)
return NonDominated
end
```

Figure 3: CSP and Pareto bi-criterion optimization

4 CASE-STUDY

4.1 Description

We propose to illustrate the contribution of CSPs method with a simple case study of mechanical component. Figure 2 is presenting a bolt coupling that is used to transmit a torque by adherence between two shafts.

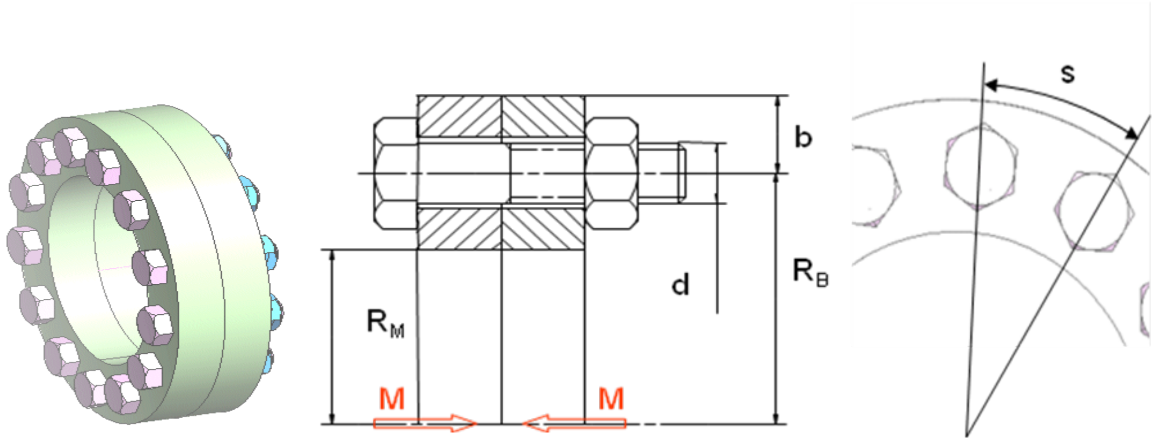

Figure 4: Bolt coupling

We suppose here that, in the design process of the product, technological choices have been made. Once designer have chosen this component, it is possible to identify design parameters {DPs} and functional requirements. In this simple case, functional requirements can be written as explicit relations between design parameters. These relations are equations and inequality relations. The model used here to establish theses relations lies on the expertise of the designer. This model is based on the VDI [13] method to calculate bolt dimension. The relations listed below will give the set of constraints for the CSP method. The mono-objective version of this problem has been previously treated in [14] with a pure genetic algorithm based approach and in [15] with a CSP one.

Table 1. Set of design parameters

Design parameters		
Geometrical parameters	d_s	diameter of stress area (mm)
	d	nominal diameter of the bolt (mm)
	b	radial width of the contact surface
	s	interval between bolt (mm)
	d_2	pitch diameter of thread (mm)
	p	pitch of thread (mm)
	s_m ,b_m	size of the tightening tools (mm)
	A_s	area of stress cross section (mm²)
	N	number of bolts
	N_m	minimal number of bolts
	R	radius of the couling (mm)
	R_b	radius of bolts (mm)
	R_m	radius of housing shafts (mm)
Functional parameters	M	torque transmitted by the coupling (N.mm)
	M_T	torque to be transmitted by the coupling (N.mm)
	F_{0mini} , F_{0maxi}	minimal and maximal tensile strength in bolts (N).
	C_1	torsion moment in the bolt due to the preload (N.mm)
	σ_{max} :	maximal normal stress in the bolt (MPa)
	τ_{max}	maximal tangential stress in the bolt (MPa)
	σeq_{max}	maximal Von Mises stress in the bolt (MPa)
	α_s	accuracy factor of the tightening tool
Material paramete	f_m , f_1	friction coefficient between rim of the coupling and threaded contact surfaces in bolt

4.2 Modeling as a constraint satisfaction problem

An analysis of the bolt coupling gives us the relations below taken as CSP constraints:

$$M = N \times R_b \times f_m \times F0_{mini} \quad (11)$$
$$F0_{maxi} = \alpha_s \times F0_{mini} \quad (12)$$
$$\sigma_{max} = \frac{F0_{maxi}}{A_s} \quad (13)$$
$$\tau_{max} = 16 \times \frac{C_1}{(\pi \times {d_s}^3)} \quad (14)$$
$$C_1 = F0_{maxi} \times (0.16 \times p + 0.583 \times d_2 \times f_1) \quad (15)$$
$$\sigma eq_{max} = \sqrt{{\sigma_{max}}^2 + 3 \times {\tau_{max}}^2} \quad (16)$$
$$s = 2 \times \pi \times \frac{R_b}{N} \quad (17)$$
$$R_b = R_m + b \quad (18)$$
$$A_s = \pi \times \frac{{d_s}^2}{4} \quad (19)$$
$$R_b \geq R_m + b \quad (20)$$
$$0.9 \times R_e \geq \sigma eq_{max} \quad (21)$$
$$M \geq M_T \quad (22)$$
$$s \geq s_m \quad (23)$$
$$b \geq b_m \quad (24)$$
$$N \geq N_m \quad (25)$$

The screw parameters set of values (Table 2) should be modeled as a constraint table.

A constraint table is a global constraint that represents the possible combination values of a set of constraint variables. By global constraint, we mean a constraint that should be propagated on complex data structures. In our case, each line of a constraint table is a tuple of consistent values. If one or several values of a constraint variable become forbidden during a CSP solving process all the tuple related to this value are removed from the table too.
For example, with Table 2, if we decide that d_2 has to be greater than 10 and p has to be different to 2 then, Lines number 1, 2, 3, 5 and 6 are removed from the table. Only lines number 4, 7 and 8 stay inside the constraint table.

Table 2. Screw parameters set of values table

Num	d	ds	d2	p	bm	sm	dt
1	6	5.062	5.350	1.00	7.50	14.50	6.6
2	8	6.827	7.188	1.25	9.50	18.50	9.0
3	10	8.593	9.026	1.50	12.50	23.50	11.0
4	12	10.358	10.863	1.75	13.50	26.50	13.5
5	14	12.124	12.701	2.00	15.50	29.50	15.5
6	16	14.124	14.701	2.00	17.00	32.00	17.5
7	20	17.655	18.376	2.50	21.00	40.00	22.0
8	24	21.185	22.051	3.00	25.00	48.00	26.0

Our CSP model has been implemented with the IlogCP C++ library [16].

We would like now to illustrate our approach with a set of numerical results. The initial domains given by the expert for the constraint variables are presented on Table 3. After a first propagation (Prune subroutine), the intervals are reduced as in the second column of Table 3. Then after adding the following specifications:

$$M_T = 4000000 \quad (26)$$
$$f_m = 0.15 \quad (27)$$
$$f_1 = 0.15 \quad (28)$$
$$\alpha_s = 1.5 \quad (29)$$
$$R_e = 627 \quad (30)$$
$$N_m = 8 \quad (31)$$
$$R_m = 50 \quad (32)$$

The next propagation step again reduces the intervals as shown in the fourth column of Table3.

4.3 Pareto bi-criterion optimization

In this case study we would like to minimize the two following functions:

- The total cost of the bolt coupling in euros :

$$C_{coupling} = K_1 \times d + K_2 \times N \quad (33)$$

$K1$ = 0.6 euros/mm (34)

$K2$ = 5 euros /bolts (35)

- The total mass of the bolt coupling ($t_h$parameter is the thickness of the coupling) :

$$M_{coupling} = \frac{\pi}{2} \times t_h \times (\rho_j \times (4 \times R_b \times b_m - N \times {d_t}^2) + \rho_v \times N \times d^2) \quad (36)$$

$$\rho_j = 2.710^{-6} \text{kg.mm}^{-3} \quad (37)$$

$$\rho_v = 7.810^{-6} \text{kg.mm}^{-3} \quad (38)$$

Table 3. CSP solving process

Design	**Initial domains**	**1st propagation**	**After specifications**
A_s	[15 , 400]	[20.12, 352.49]	[20.12, 352.49]
R_b	[5 , 1000]	[13, 1000]	[58 , 100]
R_m	[5 , 1000]	[5, 992]	50
s	[1, 100]	[14.5, 100]	[14.5 , 78.5398]
b	[5 , 50]	[8,50]	[8,50]
N	[1 , 1000]	[4 , 433]	[8 , 43]
N_m	[4 ,1000]	[4 , 433]	8
d	[6 , 24]	[6 , 24]	[6 , 24]
p	[1 , 3]	[1 , 3]	[1 , 3]
d_2	[5.35 , 22.051]	[5.35, 22.051]	[5.35, 22.051]
d_s	[5.062 , 21.185]	[5.062, 21.185]	[5.062 , 21.185]
s_m	[14.5 , 48]	[14.5 , 48]	[14.5 , 48]
b_m	[7.5 , 25]	[7.5 , 25]	[7.5 , 25]
dt	[6.6, 26]	[6.6, 26]	[6.6, 26]
M	[1000, 1e+007]	[1000, 1e+007]	[4e+006, 1e+007]
$F0_{mini}$	[10 , 100000]	[10, 66666.7]	[6201.55, 66666.7]
$F0_{maxi}$	[10 , 100000]	[15, 100000]	[9302.33, 100000]
C_1	[0 , 200000]	[2.4, 200000.]	[5840.53, 200000]
σ_{max}	[0 , 2000]	[0.0425, 1110.6]	[26.3903, 564.274]
τ_{max}	[0 , 2000]	[0.00128, 641.2]	[3.12851, 325.442]
σeq_{max}	[0 , 2000]	[0.0426, 1110.6]	[26.9409, 564.3]
α_s	[1.5 , 4]	[1.5 , 4]	1.5
f_m	[0 , 1]	[3.464e-008, 1]	0.15
f_1	[0 , 1]	[0 , 1]	0.15
R	[10, 1050]	[21, 1050]	[66, 150]

The goal is to minimize both $C_{coupling}$ and $M_{coupling}$. To do that, we implement the BiPareto algorithm and obtain the Pareto frontier of the problem (Figure 5). We notice that in our test case the Pareto frontier is discontinuous and not convex.

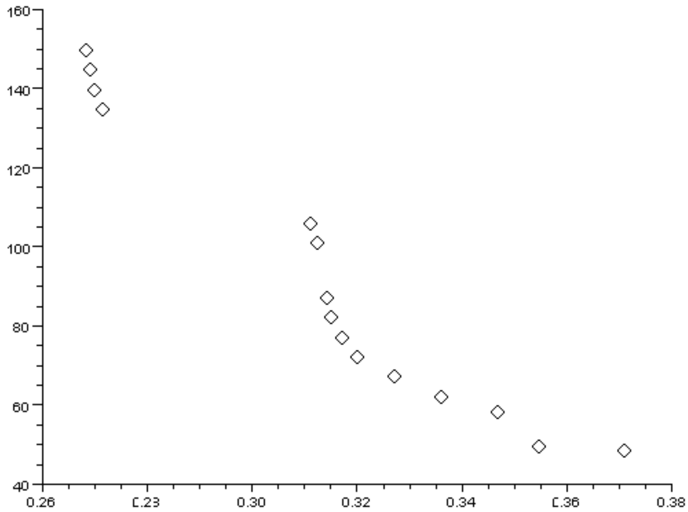

Figure 5: Pareto frontier of the bolt coupling sizing problem

5 CONCLUSION

As a conclusion, we can outline that CSP method are able to support the phase of reducing the size and the complexity of the design space. CSP modelling and solving processes avoid complex analytical manipulation of function requirements. Designer can test several ways of constraining the design by adding or deleting constraints. This incremental process is very useful in this phase of reducing the design space, in an integrated and collaborative design process.

For this simple example, the BiPareto algorithm finds easily the Pareto frontier optimal solution. One of the advantage is that designers don't have to simplify the formulation of the optimization problem. Another one is that our proposal allows to deal with all types of bi-dimensional Pareto frontier even if they are discontinuous or non convex.

Our current research concerns the generalization of this approach to an n dimensional Pareto Surface algorithm based on the CSP mechanisms.

REFERENCES

[1] Gupta S., Tiwari R., Nair S.B., Multi-objective design optimization of rolling bearings using genetic algorithms, *Mechanism and Machine Theory, Volume 42, Issue 10,* October 2007, pp 1418-1443

[2] Montanari U., Networks of constraints: fundamental properties and applications to picture processing, *Information Science* 7, 1974, pp.95-132.

[3] Mackworth A.K., Consistency in networks of relations, *Artificial Intelligence* 8, 1, 1977, pp.99-118.

[4] Moore R.E., *Interval Analysis, Prentice-Hall*, 1966.

[5] Davis E., Constraint propagation with interval labels, Artificial Intelligence, v. 24.3, 1987.

[6] Falting B., Arc consistency for continuous variables, Artificial Intelligence 65(2), 1994.

[7] Benhamou F., Granvilliers L., *Continuous and Interval Constraints. In Handbook of Constraint Programming,* Chapter 16:571-604, 2006.

[8] Vargas C., Saucier A., Albert P., Yvars P.A., Knowledge Modelisation and Constraint Propagation in a Computer Aided Design System, *In Workshop notes Constraint Processing in CAD of the Third International Conference on Artificial Intelligence in Design*, Lausanne, Switzerland, August, 1994.

[9] Yannou B., Harmel G., Use of Constraint Programming for Design, in Advances in Design, El Maraghy W. Editors, Springer, chapter 12, 2005.

[10] Chenouard R., Sebastian P., Granvilliers L., Solving an Air Conditioning Problem in an Embodiment Design Context using Constraint Satisfaction Techniques, CP'2007, 13th International Conference on Principles and Practice of Constraint Programming, 2007.

[11] Yvars, P.A., 2008, Using constraint satisfaction for designing mechanical system, *International Journal on Interactive Design and Manufacturing (IJIDeM),* Volume 2, Number 3, August 2008.

[12] L. N. V. Wassenhove and L. F. Gelders. Solving a bicriterion scheduling problem. *European Journal of Operational Research*, 4(1):42-48, 1980.

[13] V.D.I.2230, *Systematic Calculation of High Duty Bolted Joints*, 1998.

[14] Giraud, L., Lafon, P., Optimization of mechanical design problems with genetic algorithms, *Proceeding of the 2nd International Conference IDMME'98,* Compiègne, France, pp. 90-98, 1998.

[15] Yvars P.A., LAfon P., Zimmer L., Optimization of mechanical system: Contribution of constraint satisfaction method, *proc of IEEE Computers and Industrial Engineering CIE'39*, Troyes, France, 2009.

[16] Ilog, *IlogCP, Reference Manual, Ilog*, Gentilly, France, 2009.

Contact: P.A. Yvars
Institut Supérieur de Mécanique de Paris (SupMeca)
LISMMA
3 rue Fernand Hainaut
93407, Saint Ouen Cedex
France
Phone : int +33(0)1 49 45 29 25
e-mail : pierre-alain.yvars@supmeca.fr
URL : http://lismma.supmeca.fr

Pierre-Alain Yvars is an Associate Professor at the Institut Supérieur de Mécanique, Paris. His current subjects of research focus on the declarative meta modeling of complex systems, design processes, and production systems, the application of constraint programming techniques and interval propagation to solve and optimize design problems.