A constraint based approach for aiding heat treatment operation design and distortion evaluation

M. Aldanondo[1], E. Vareilles[1], K. Hadj-Hamou [2] and Paul Gaborit [1]

1 Centre de Génie Industriel - Ecole des Mines d'Albi-Carmaux
Campus Jarlard - 81013 Albi CT Cedex 09 - France
{aldanondo , vareilles, gaborit}@enstimac.fr
2 GILCO - ENSGI – INP Grenoble - France
hamou@gilco.inpg.fr

Abstract. This paper presents an interactive constraint based system that simultaneously assist design and evaluation. This work is driven by an industrial case dealing with heat treatment operation. The first part presents the problem and provide ideas of the solution. Then the knowledge model mixing discrete and numerical constraints is presented. The third section provides filtering elements in order to permit interactive assistance. The last one discusses the designed system. The originality of the proposition lies in the gathering of classical discrete constraints filtering techniques with numerical constraint 2B consistence filtering mechanisms that were necessary to respond to the industrial need.

1   Introduction

The goal of this communication is to present an interactive constraint based system that simultaneously allows interactive design of a heat treatment operation and qualitative distortion evaluation. This problem originates from a European project called VHT for "Virtual Heat Treatment" (project No G1RD-CT-2002-00835).

A heat treatment operation consists in raising the temperature of a steel part until a certain temperature, keeping at this temperature for a while, then cooling down rapidly. The expected effect is an improvement of the mechanical properties of the part. But, simultaneously, a negative effect corresponding with part distortions occurs most of the time.

A heat treatment operation can be defined according to :
- the material of the part, characterized by parameters relevant to the chemical composition and the mechanical/thermal/structural behaviours,

- the geometry of the part, characterized by parameters relevant to the shape, massivity, symmetry of the part, hole existence...,
- the heat treatment conditions, characterized by parameters describing the heating device, the cooling system, the parts layout during heating and quenching and various tuning (duration, temperature, flow-rate...).

Classical heat treatment knowledge gathers relations between previous parameters permiting the definition of a heat treatment operation which improves the mechanical properties of the part. In order to estimate distortion, some specific knowledge can be added in order to give some kind of a qualitative score.

The goal of this paper is to show that these two kinds of knowledge can be formalized as a constraint satisfaction problem (CSP) and permit to provide an interactive assistance tool for defining heat treatment operations and evaluating relevant distortion. The rest of the paper is organized as follows.

The second section presents the knowledge model gathering a design part and an evaluation part. It will be shown that three kinds of constraints are necessary: discrete constraints, numerical constraints and mixed constraints. As we target an interactive assistance, the filtering elements for each kind of constraints will be described in section three. Then some interests and limits of the system will be discussed in the last section.

2 The knowledge model

The goal of this section is to present an overview of the knowledge model. This section is divided in two parts. The first one describes the model piece relevant to design while the second deals with the evaluation part. The last one shows how the two model pieces fit together.

For each sub section, the model is described as a constraint satisfaction problem (CSP) [1] and outlines the kind of variables and the kind of constraints that are necessary. Various models were designed with heat treatment experts, coming both from academia and industry, through a dozen of meetings planed during the past18 months [2]. The most advance one concerns parts that belong to a part family "axis" (one dimension much longer than the two others) and is considered in this paper.

2.1 Piece of model relevant to heat treatment design

Around 50 parameters have been identified by the experts. Each parameter is associated with a design variable {vp} upon which the user can input a domain restriction. These variables are either symbolic or numerical. Two kinds of constraints where used.

2.1.1 Compatibility constraints
Most of the constraints relevant to heat treatment design are compatibility constraints expressing restrictions on combinations of variable values. As variables are either symbolic or numerical, three kinds of compatibility constraints are present, but all of

them can be represented thanks to compatibility tables. The three kinds of constraints are :
- discrete : when all constraint variables are symbolic. In this case constraints represents allowed combinations of symbolic values.
- continuous : when all constraints variables are numerical. In this case constraints represents allowed combinations of intervals,
- mixed : when constraint variables are symbolic and numerical. In this case constraints represents allowed combinations of intervals and symbolic values.

2.1.2 Activity constraints
In order to be able to modulate the existence of some parameters, it is necessary to be able to express some variable existence conditions. This kind of constraints, called activity constraint in the dynamic extension of CSP [3], have the following shape, X = "x" => Y exists, meaning if variable "X" equal the value "x" then variable "Y" exists. They have been mainly used to describe geometric parameters of the part as for example :
- hole_existence = "yes" => Variable_hole_diameter exists,
- part_thickness $\in$ [10 , 20] => Variable_part_area exists.

2.2   Piece of model relevant to the evaluation of distortion

As part distortion is geometrically very complicated to describe, heat treatment experts have proposed to quantify five basic distortion components for the part family "axis": (i) "spool/barrel", (ii) "banana", (iii) "ovalization", (iv) "spacing/tightening" and (v) "umbrella". It can be noted that "spacing/tightening" distortion can exist, only if the part has got a hole and "umbrella" distortion can exist if shoulder shapes are present. Previous activity constraints are necessary to model this two conditions.

In the CSP framework, a way to compute the evaluation of a solution is to assign weights to each parameter values and to each allowed combinations of parameter values relevant to constraints. Because the goodness of a solution is usually influenced by both the quality of the values chosen and the resulting associations of values. The evaluation of a solution is defined as the sum of the weights of all the values and pairs of values involved in the solution [4]. The main interest of this approach lies in its simplicity and its optimisation possibilities. But in an interactive design process, this approach does not easily allow the user to input some constraint on the resulting score and to see the consequences on the parameter values. In our case, the possibility to input a maximum value on a distortion component and to input parameter values with respect to this constraint is a necessity. Therefore, it was decided to use a simple formulae to compute the distortion score relevant to each distortion component. Next sub sections present necessary evaluation variables and evaluation constraints.

2.2.1 Evaluation variables
In order to calculate such a score, heat treatment experts propose the following approach based on three kinds of distortion attributes:

- a first small subset (15 parameters) of heat treatment parameters, vp, allows a first quantification of each distortion component, named potential distortion attribute, vd_pot,
- a second larger subset (45 parameters) of heat treatment parameters, vp, permits to quantify intermediate distortion attributes, vd_int. These attributes modulate each potential distortion component in a similar way. 26 intermediate distortion attributes have been identified.
- the resulting score of each distortion component, named final distortion attribute vd_fin, is computed as a product of the relevant potential distortion attribute and the set of intermediate distortion attributes.

Each distortion attribute is associated with an evaluation variable. These variables are all numerical and defined within intervals. In order to compare and to normalize solutions, heat treatment experts decided that (i) each final distortion component (vd_fin) is quantified into an interval [1, 1000] , where "1" means no distortion and "1000" means maximum distortion, (ii) each potential distortion component (vd_pot) is quantified into an interval [1, 20] while (iii) each intermediate distortion attribute (vd_int$_k$) is quantified into intervals [1, α_k] such their product belongs to an interval [1, 50].

2.2.2 Evaluation constraints

As potential and intermediate distortion attributes are all numerical and parameters either symbolic or numerical, constraints linking these distortion attributes with parameters are either mixed or numerical. All these constraints are expressed thanks to compatibility tables : (i) 5 constraints for the quantification of the 5 components of potential distortion (vp, vd_pot) and (ii) 26 constraints for the quantification of the 26 intermediate distortion attributes (vp, vd_int).

As each final distortion component is obtained with the computation of the product of the relevant potential distortion component and the set of intermediate distortion attributes, the following numerical constraint : vd_fin = vd_pot*$\Pi_{k=1 \text{ to } 26}$(vd_int$_k$) must be defined for each of the five distortion components.

2.3   Gathering the two model pieces in a single model

The two model pieces are gathered in a single design/evaluation model as shown in figure 1 with two distortion components. This model shows :
- two kinds of variable (represented with circles on figure 1) associated with :
    - parameters (vp), symbolic or numerical variables,
    - distortion attributes (vd_pot, vd_int, vd_fin), numerical variables,
- three kinds of constraints between :
    - parameters (vp), these constraints gathers activity and compatibility constraints, they can be symbolic, numerical or mixed constraints, they are mainly expressed thanks to compatibility tables (solid lines on figure 1), they correspond with the knowledge relevant to heat treatment operation,
    - parameter (vp) and distortion attributes (vd_pot, vd_int), these constraints are compatibility constraints, they can be numerical or mixed constraints, they are described with compatibility tables, (doted lines on figure 1), they correspond

with the knowledge relevant to the distortion evaluation of heat treatment operation,

- distortion attributes (vd_pot, vd_int, vd_fin), these constraints are compatibility constraints, they are numerical constraints, they are defined with the "product" mathematical operation (broken lines on figure 1).
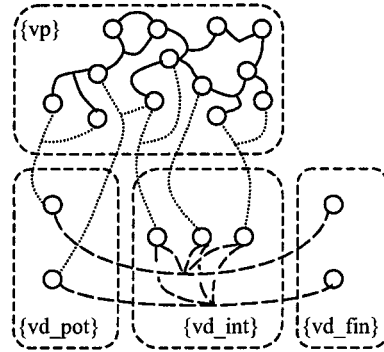
Fig.1   Architecture of the model

The object of the next section is to study how this model can be used to interactively and simultaneously allows design and evaluation of a heat treatment operation.

3   Filtering techniques

The object of this section is to present the various filtering techniques that are necessary and to show how they are gathered in a single filtering engine.

3.1   Compatibility constraints

For discrete constraints, the simple arc consistency technique (AC-3) is used to propagate this kind of constraints. As some constraints have an arity larger than two, AC-3 has been adapted.

For mixed and numerical constraints expressed with compatibility tables including intervals, the association of a label with each interval permits to consider them as discrete constraints [5]. The previous filtering means can be used except that when the definition domain of a numerical variable is reduced, it is necessary to reconstruct the definition domain of the variable with classical set operations (union, intersection). Filtering therefore goes as follows:

    1 - Input on variable var-x
    2 - Put var-x in list-var-1
    3 - While list-var-1 non empty :
    4 - Take variable var-i of list-var-1
    5 - Put constraint including var-i in list-cst-1
    6 - While list-cst-1 non empty :
        7 - Take cst-j of list-cst-1
        8 - Filter cst-j with var-i
        9 - If a reduced var is numerical : Reconstruct definition domain of the reduced var
        10 - Put var, with a reduced domain, in list-var-1

When the variables are all numerical and constraints written with a mathematical formula, $f(x_1, x_2...x_n) = 0$, 2B-Consistency, proposed by [6] and based on interval arithmetic [7], proposes filtering techniques that operate fine if :

(i) $f(x_1, x_2...x_n) = 0$ can be projected on any variable x_i , meaning that a function f_i exists as : $x_i = f_i(x_1, x_2....x_{i-1}, x_{i+1},...x_n)$

(ii) any projection f_i is continuous and monotonous.

(iii) only one constraint expressed as a formula acts on a same variable subset. 2B-Consistency is weak when more than one constraint acts on a same sub-set of variables (corresponding to some constraint intersection). It is shown in [8] that a simple problem gathering two variables and three constraints cannot be fully filtered.

(iv) each variable occurs only one time in a formula. In the opposite case, for example: $x_1^2 - x_1 - x_2 = 0$, it has been shown that the way to express the mathematical expression, for example : $x_1^2 - x_1 - x_2 = 0$ or $x_1*(x_1 - 1) - x_2 = 0$, influences the quality of the filtering operation [8].

As this kind of constraints is only present in our model for computing the product quantifying the final distortion attribute (vd_fin = vd_pot * $\Pi_{1 \text{ to } 26}$(vd_int)), the previous restrictions are not present and 2B-Consistency can be used. For a single continuous constraint, filtering is done as follows :

1- Input on variable var-x
2- Put var-x in list-var-1
3- While list-var-1 non empty :
4- Take variable var-i of list-var-1
5- Put all constraint projections including var-i in list-cst-1 except the one relevant to var-i
6- While list-cst-1 non empty :
 7- Take constraint projections of list-cst-1
 8- Compute the resulting domain of the projected variable
 9- Intersect resulting domain with initial domain of the projected variable
 10-If domain is reduced, put projected variable in list-var-1

3.2 Activity constraints

In order to deal with activity constraints, we only need to use the "Require Variable" activity constraint (var-x = "x" => var-y exists) among the four categories (Require, Require not, Always require, Always require not) proposed by [3] Resulting filtering goes as follows :

1- Input on variable var-x
2- Put var-x in list-var-1
3- While list-var-1 non empty :
 4- Take variable var-i of list-var-1
 5- Put activity constraint including var-i in its premise (left part of =>) in list-a-cst-1
 6- While list-a-cst-1 non empty :
  7- Take a-cst-j of list-a-cst-1
  8- Evaluate the premise of a-cst-j
  9- If true, add triggered var-y (right part of =>) in the current problem and put this variable in list-var-1

3.3   Gathering the filtering techniques in a single engine

The three filtering techniques have been gathered in a single engine. It was decided to first filter activity constraints, then constraint expressed with compatibility tables to finish by constraints defined by formulae. The architecture of the global filtering is therefore as follows:

> Input on var
> var in list-var-1
> While list-var-1 non empty
> > Take variable var-i of list-var-1
> > *Filter activity constraint (section 3.2)*
> > Result : can add variables to the current problem and put them in list-var-1
> > *Filter discrete, mixed, continuous constraints defined with compatibility tables (section 3.1)*
> > Result : put variables with a reduced domain in list-var-1
> > *Filter Continuous constraints defined with formulae(section 3.1)*
> > Result : put variables with a reduced domain in list-var-1

4   Interest and limits of the proposed approach

All these elements have been set up in a software mock-up that can be seen on the web at : http://iena.enstimac.fr:20000/cgi-bin/vht.pl. Two modes to use the system have been identified in the beginning of section 2.2. The first one, consisting in interactively inputting restrictions only on parameters and see the computation of the relevant distortion, raises an interesting point. While the other, inputting parameter restrictions (that are not negotiable) and some threshold on the maximum value of the distortion in order to get domain restrictions (on negotiable parameters), points out a limit of the approach.

For the first mode, at the beginning of the design process, each final distortion attribute has a definition domain equal to the interval [1, 1000] meaning that no information about distortion level is available. As filtering is launched after each user input, each final distortion domain is progressively reduced during the design process. Even if sometime this definition interval is split in different intervals, this allows the user to see the progressive effects of his selections on each final distortion component. This interesting behaviour is allowed by 2B-Consistency that operates with interval arithmetic. At the end of the design process, each final distortion attribute has a domain corresponding with a reduced interval. An average between the lower and upper interval bounds can be calculated for solution comparison.

The previous good point has a drawback while dealing with the second mode. If the user inputs restrictions that affect variables corresponding with parameters (vp) and final distortion attributes (vd_fin), the proposed filtering approach can lead to an inconsistent problem. This problem comes from the arc consistence filtering techniques that consider sequentially only one constraint at a time and check only the consistence of pair of variables. Filtering with a stronger consistency could avoid this problem but would be too much time consuming.

5   Conclusions

The goal of this communication was to present an interactive constraint based system that simultaneously allows interactive design and qualitative evaluation of heat treatment operations.

A knowledge model gathering a design model piece and an evaluation model piece was designed. The corresponding constraint satisfaction problem uses discrete and numerical variables, discrete, mixed and numerical constraints. Classical filtering techniques, for discrete constraints (compatibility tables), have been embedded with 2B-consistence, for numerical constraints (formulae), in a single filtering engine.

The resulting configuration software allows the user to input restrictions either on the parameters defining the operation to design or on the attributes that characterize the evaluation score.

The model and aiding design software are now in a validation step performed by end users. In spite of the aiding decision tool, users underline the fact that they better understand heat treatment process. The drawback, discussed in section 4, is avoided during design by progressively reducing final distortion attributes with a "try and error" procedure.

In terms of knowledge modelling, it can be pointed out that heat treatment experts succeeded rather quickly to deal with the formalism variable/domain/constraint. This confirms, as frequently reported in other papers, that the natural and easy to understand concept of the CSP approach makes it a good candidate for domain knowledge representation.

References

1. Tsang E.: Foundations of constraints satisfaction. Academic Press London (1993).
2. David Ph, Veaux M., Vareilles E., Maury J.:Virtual Heat Treatment Tool for Monitoring and Optimising Heat Treatment Process.   2e International Conference on Thermal Process Modelling and Computer Simulation, Nancy, (2003).
3. Mittal S., Falkenhainer B.: Dynamic Constraint Satisfaction Problems. AAAI ConferenceBoston USA, (1990) 25-32.
4. Hulubei T., Freuder E.C., Wallace R.J.:. The Goldilocks problem. AI EDAM, Volume 17, Number 1, (2003) 3-11.
5. Faltings B.: Arc consistency for continuous variables. Artificial Intelligence, volume 65, (1994) 363-376.
6. Lhomme O. : Consistency techniques for numerical CSPs. IJCAI conference, Chambéry, France, (1993).
7. Moore R.E.: Intervals Analysis. Prentice Hall, (1966).
8. Lhomme O., Rueher M. : Application des techniques CSP au raisonnement sur les intervalles. Revue d'intelligence artificielle, Dunod, Vol. 11, (1997) 283-311.