



HAL
open science

Search Heuristics for Constraints-Aided Design

Raphael Chenouard, Laurent Granvilliers, Patrick Sébastian

► **To cite this version:**

Raphael Chenouard, Laurent Granvilliers, Patrick Sébastian. Search Heuristics for Constraints-Aided Design. AI EDAM, 2009, 23 (2), pp.175-195. 10.1017/S0890060409000055 . hal-00426706

HAL Id: hal-00426706

<https://hal.science/hal-00426706>

Submitted on 23 Mar 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Search Heuristics for Constraint-Aided Embodiment Design

R. Chenouard ^{a,*}, L. Granvilliers ^b P. Sébastian ^a

^a*ENSAM Bordeaux, TRansferts Ecoulements FLuides Energétique, CNRS,
F-33405, Talence Cedex*

^b*University of Nantes, Laboratoire d'Informatique de Nantes Atlantique, CNRS,
BP 92208, F-44322 Nantes Cedex 3*

Abstract

Embodiment Design (ED) is an early phase of product development. ED problems consist of finding solution principles that satisfy product requirements such as physics behaviors and interactions between components. Constraint satisfaction techniques are useful to solve constraint-based models that are often partial, heterogeneous, and uncertain in ED. In this paper, new constraint satisfaction techniques are proposed to tackle piecewise defined physics phenomena or skill-based rules and multiple categories of variables arising in design applications. New search heuristics and a global piecewise constraint are introduced in the branch-and-prune framework. The capabilities of these techniques are illustrated on both academic and real-world problems. The latter have complete models presented in the appendix.

Key words: search, heuristic, embodiment design, constraint satisfaction

* Corresponding author.

Email addresses: `raphael.chenouard@bordeaux.ensam.fr` (R. Chenouard),
`laurent.granvilliers@univ-nantes.fr` (L. Granvilliers),
`patrick.sebastian@bordeaux.ensam.fr` (P. Sébastien).

1 Introduction

The design process is a sequence of phases from the definition of needs and requirements to preliminary design and detailed design (Pahl & Beitz, 1996). Preliminary design includes conceptual design (CD) and embodiment design (ED). The ED phase investigates the feasibility of some product schemes obtained from the CD phase. This phase mainly tackles physics behaviors and interactions between the product, its components and environments. Product modeling is based on the definition of the laws of physics, functional models, economic criteria, etc.

In this paper we focus on robust ED taking into account variability, uncertainty or imprecision in the design process. The goal is to determine the main structuring characteristics of a product, such as the working structure, standard components, and the main dimensions, while no significant decisions have been taken at that point. Several components may change during this phase. Robust ED can be implemented in a constraint-based approach. Product models can be translated into numerical constraints. Uncertainty and imprecision can be partially captured by interval computations. Heterogeneous models and incomplete information can naturally be dealt with. These models are involved in robust design approaches taking into account mathematical models during the early phases of design process. Robustness may be regarded through its meaning within the design community (Rothwell & Gardiner, 1990).

Product modeling leads to the definition of several types of constraints. Product behavior laws relating to physics analysis are expressed through conservation law, which are easily translated into constraints. In some cases, behavior

laws are defined by sets of phenomenological relations, namely piecewise relations depending on one or several parameters. Product modeling also leads to the definition of several types of variables. *Design variables* are related to the main dimensions and characteristics of the product. Designers are interested in finding out powerful solution principles, where design variable values correspond to high performance criteria. Performance criteria may be represented by *Performance variables*. Other variables of the model are *Auxiliary variables*, maintained within the model to link design variables to performance variables in order to preserve the model intelligibility. They are introduced by the modeling phase (see Figure 1).

In this figure, the embodiment design knowledge of a product takes into account design variables, whose values identify each design solution. Designers use also several criteria to observe and evaluate the design solutions. Several diagrams and charts are used to identify the product functions and decomposition (technical organization charts) and to investigate the physics phenomena regarding fluxes and induced effects (fluxes flow diagram and substance field graph). In the modeling part of the embodiment design phase, these concepts are translated into a mathematical representation. Obviously, the variables already defined in the knowledge representation are also in the mathematical model, and in most cases, criteria are easily expressed with constraints and some variables to observe criteria values. But diagrams and charts must be converted in a computable form. New variables are introduced and they do not correspond to designers' decision parameters. Thus, these new variables and constraints describe physics phenomena, products geometry characteristics, etc. Some functional variables are defined to preserve the model intelligibility

and to express, for instance, well-known physics dimensionless numbers characterizing physics phenomena. Some of them are introduced after some steps of model reductions.

Our purpose is to define new constraint satisfaction techniques in the interval-based branch-and-prune framework to solve enriched models of ED applications. We investigate enriched robust ED models, since we consider various knowledge about products: specifications and requirements, knowledge of designers concerning the whole products life cycle, physics phenomena, etc. All this knowledge is required to compute quite safe and robust values (from a design point of view) for the main variables of an ED model. The first problem is to handle specific physics phenomena. To this end a global piecewise constraint is defined at the modeling and the solving levels. The second problem is to tackle the different types of variables. Existential quantifiers are introduced in the constraint-based model to take into account the fact that auxiliary variables are meaningless from a design point of view. New heuristics allow the differentiation of the variables during search, according to their types. An experimental study from a prototype and several benchmarks are reported. Complete models are given in appendix for people who want to make their own test with real world applications.

Section 2 introduces CSP modeling for ED. Solving principles are presented and some search strategies are stated in Section 3. Experimentations on academic problems are carried out in Section 4. ED models derived from existing engineering models are processed in Section 5. Our approach is compared to some related work in Section 6.

2 Problem Modeling

We consider embodiment design problems defined as mixed models including integer variables, real variables, constraints and piecewise constraints. The main idea of this paper is to distinguish between variables according to application requirements and to separate them in several sets during the search phase of solutions. A model is defined by a set X of variables lying in some domain D and a set of constraints C . Each constraint is a restriction of D given atomic formula over the usual structure of real numbers. Our goal is to find values in D for the variables of X satisfying all the constraints in C .

2.1 Types of Variables

In ED problems, two types of variables are highlighted: the auxiliary variables, and the main variables including the design variables and the performance variables. The main variables must be computed at a given discernment precision. The values of the auxiliary variables may be useless from the designer's point of view, no initial precision or carefully chosen precision may be defined. The distinction between main variables and auxiliary variables is always possible, since main variables are stated by the product specifications and requirements. The main variables are shared by all design phases. They identify the main characteristics of products, that's why their domains and precisions are well known on the contrary to the auxiliary variables, which are specific to each design phases.

Notations: Given a variable or a set of variables x , a real number or a set of real numbers r and a constraint or a conjunction of constraints C on x , we

write $C(r)$ if C is satisfied when x has value r .

Let $X = (x_1, \dots, x_n)$ denote the main variables, let $Y = (y_1, \dots, y_m)$ denote the auxiliary variables, and let D_X and D_Y be their domain. To solve ED problems may be seen as the computation of the set of solutions on the main variables, where there is at least one solution for auxiliary variables:

$$\{r_X \in D_X | \exists r_Y \in D_Y \wedge C(r_X, r_Y)\} \quad (1)$$

where C stands for the constraints to be satisfied.

In other words, the main variables define a scheme of solution for designers, namely, the main architectures of a product. The descriptions of the product and its components concerning their behavior, geometry, etc. make these architectures physically valid, if at least one solution is found on the auxiliary variables for each architecture.

Several approaches can be used to tackle such problems. Search problems may correspond to our ED problems, since solutions to ED problems are other than yes or no, contrary to decision problems. But it can be seen in Beame et al. (1995) that for each search problem an equivalent decision problem exists and in an ED context, it may be expressed as:

$$\{\exists r \in D | C(r)\} \quad (2)$$

where all variables are linked with an existential quantifier. Efficient SAT algorithms (Cook & Mitchell, 1997) can be used in this case, but since an existential quantifier is applied to each variable, only one solution may be found to be the yes answer.

The constraint satisfaction problem (CSP) approach defines a framework for solving general problems expressed as a conjunction of constraints, where all variables are free:

$$\{r \in D | C(r)\} \tag{3}$$

All values r for variables satisfying C are computed. This approach does not match the formulation in (1), but the solving algorithms can be adjusted to undertake an existential quantifier on some variables.

We implement our approach and its corresponding algorithms within a CSP framework that uses continuous domains. This framework is suitable for the ED problems (Zimmer & Zablit, 2001; Gelle & Faltings, 2003; Vareilles et al., 2005). The CSP approach allows designers to make their models evolve very quickly as opposed to other methods, where designers express the knowledge, while carrying out its coding related to numerical solving methods similar to the constraint satisfaction approach. Some examples based on an evolutionary approach may be found in Sébastien et al. (2006). Moreover, the solving process of a CSP guarantees the completeness of the set of approximate solutions, whereas other methods are often linked with relaxations and approximations of some stochastic solutions.

2.2 Intervals computations and variable precision

The problem of computing solutions for functions on real numbers is known to be undecidable (Richardson, 1968; Wang, 1974). Computers arithmetic (see IEEE754 standard) defines a subset of real numbers, called the floating-point numbers. Without any other techniques, computations are made on the

floating-point numbers set and rounding errors may be important after several computation steps.

Interval arithmetic (Moore, 1966) guarantees safe computations using floating-point numbers as interval bounds. For each real number a , an interval $\mathit{hull}(a) = [a^-, a^+]$ may be used, corresponding to the smallest interval including it, where a^- is the highest floating-point number smaller than a and a^+ is the lowest floating-point number higher than a . Furthermore every operator and function must be extended from real numbers to intervals with real bounds and then a hull with floating-point bounds may be computed. For example, the three basic operators on real numbers can be extended as follows:

$$[a, b] + [c, d] = \mathit{hull}([a + c, b + d]),$$

$$[a, b] - [c, d] = \mathit{hull}([a - d, b - c]) \text{ and}$$

$$[a, b] \cdot [c, d] = \mathit{hull}([\min(a \cdot c, a \cdot d, b \cdot c, b \cdot d), \max(a \cdot c, a \cdot d, b \cdot c, b \cdot d)]),$$

where $\mathit{hull}([a, b]) = [a^-, b^+]$ and a^- and b^+ are the closest floating-point numbers lower than a and upper than b .

Other notations: Given a variable x , an interval I and a constraint C on x , we write $C(I)$ if C is satisfied in the interval sense when x takes value I . The size of an interval $I = [a, b]$ is equal to $w(I) = b - a$. Given a set of real numbers A , the hull of A , denoted by $\mathit{hull}(A)$, is the smallest interval enclosing A .

Real values in intervals cannot be enumerated as discrete domains, but intervals are split to reduce their width since a smallest hull is computed or an interval precision is reached. A precision $p(x)$ may be defined for a variable x .

It defines the interval width, where we do not want any more computations to be done. The precision on variables domain allow designers to define the tolerance authorized on some important variables, like the main variables of an ED model. Auxiliary variable precisions may be difficult to agree on, It must be highlighted that the set of auxiliary variables is often under-constrained, since, in the ED phase, some uncertainties remain about some product characteristics and its behavior.since physics phenomena are often complex. Two types of precisions may be highlighted in ED. The precision on main variables corresponds to the precision of discernment of design architectures, whereas precisions on auxiliary variables define numerical precisions for computations. To define precisions on all types of variables may increase the efficiency of the computing process, since an interval precision is often achieved before the smallest hull (or canonical hull) of a real number.

Suppose that $p(x_k) \geq 0$ (the value 0 for a precision expresses the need of a canonical interval box for a variable) is the desired precision of x_k ($k = 1, \dots, n$). We now consider the finite set of approximate solutions:

$$\{I \subseteq D_X | \exists J \subseteq D_Y \wedge C(I, J)\} \quad (4)$$

where $I = I_1 \times \dots \times I_n$ and $J = J_1 \times \dots \times J_m$, such that I is precise enough, i.e., $w(I_k) \leq p(x_k)$ for $k = 1, \dots, n$ and each interval bounds are floating-point numbers. The first goal is to compute a subset of (4) enclosing (1) having a minimal cardinal. To this end the main variable values must be close to their precisions, i.e., $w(I_k) \approx p(x_k)$. The second goal is to prove the existence of a solution (element from set 1) in every resulting box. Proofs of existence can be implemented by interval analysis techniques and this will be detailed in the next section.

2.3 CSP notions

A CSP is defined by three sets corresponding to a set X of variables, a set D corresponding to their domains and a set C of constraints restricting the variables values. The goal is to find every element of D that satisfies all constraints at the same time. This problem is unsolvable given continuous domains and transcendent functions. A more practical goal is to compute a finite approximation of the set of solutions (Lhomme, 1993). The most common approach is to calculate a set of interval boxes of a given size enclosing the solution set.

The satisfaction of a numerical constraint is usually defined as follows: every variable is interval-valued, every expression is evaluated using interval arithmetic (Moore, 1966), and every relation between intervals is true whenever there exist reals within intervals that satisfy the following relation:

$$\{r \in D_X : c(r) \rightarrow C(\text{hull}(r))\}, \quad (5)$$

where C is the interval extension of the constraint c on reals (i.e.: each variable is replaced by its interval domain and each function or operator is extended to the interval arithmetic).

The satisfaction of constraints is verified using consistency techniques. Variables' domains are checked considering the whole constraints set. If a domain is not consistent, then all unauthorized values (or intervals) are removed as long as they do not satisfy at least one constraint. Applying a global consistency is, in general, too expensive. Thus, local consistency algorithms, such as 2B and 3B-consistency (Lhomme, 1993) and box-consistency (Benhamou et al., 1999), are used instead. For instance we can consider the following defini-

tion of box consistency:

Given C the interval extension of a constraint c on reals and a box of interval domains $I_1 \times \dots \times I_n$, c is satisfied according to box consistency, if for each k in $\{1, \dots, n\}$:

$$I_k = \{a_k \in I_k \mid C(I_1, \dots, I_{k-1}, \text{hull}(a_k), I_{k+1}, \dots, I_n)\} \quad (6)$$

As soon as there are several solutions, consistency techniques are no longer sufficient. Search algorithms are used to explore the totality of the search space. Typically, a domain is chosen and is split into two disjoint intervals using a bisection algorithm. Then two new smaller problems are solved with the same iterative approach. The union of these two problems is equal to the initial CSP which is finally split in many sub-problems. The choice of the domain to split may take into account heuristics in order to optimize the search phase (for instance: most constrained variables, greatest domain, smallest domain, etc.). Then several algorithms may be used to explore such hierarchy of problems like generate and test, backtrack search, back jumping, dynamic backtracking, etc (Rossi et al., 2006).

Interval solvers implement branch-and-prune techniques (Hyvönen, 1989). The search space is given by an interval box that is iteratively split and reduced using a fixed-point approach to guarantee that the solving process converges. Efficient pruning algorithms merge consistency techniques and numerical methods. In general, splits for real variables are based on bisection, whereas integer variables are enumerated. Let us point out that integer variables can be processed as real variables within interval pruning methods and further refined using the integrality condition.

2.4 Piecewise constraint

We consider a new type of constraints for modeling piecewise defined physics phenomena. Behavior laws defining complex phenomena are often established by experiments. These experiments are done under several hypotheses and conditions defining the contexts of use for these stated laws. In many cases, the main context of experiments can be managed by only one parameter, which values identify the relation to apply. For instance, many models in fluid mechanics involve the Reynolds dimensionless number. The Reynolds number value points to different types of fluid flowing (laminar, transient, turbulent) corresponding to fluid mechanics laws (see Figure 2).

Let this constraint be

$$\text{Piecewise}(\alpha, I_1 \rightarrow C_1, \dots, I_p \rightarrow C_p)$$

Such that α is a variable, each I_k is an interval, and each C_k is a constraint or a set of constraints. The I_k identify the different cases of the piecewise phenomenon considering the parameter α and the C_k correspond to the relations to use. The intersection $I_j \cap I_k$ must be empty for every $j \neq k$, otherwise at least two constraints will apply for the same phenomenon. In other words, all the I_k define a partition of the domain of α . The piecewise constraint is satisfied if:

$$\exists k \in [1..p], D_\alpha \subseteq I_k \wedge C_k \tag{7}$$

The piecewise constraint is equivalent to C_k whenever α belongs to I_k . At most one k must exist since the I_k do not intersect, otherwise several constraints

are taken into account, which lead to an inconsistent set of constraints.

Interval constraint satisfaction techniques are used to reduce variable domains.

Let D_α be the domain of α . Four cases can be identified:

1. If a k exists such that $D_\alpha \subseteq I_k$ then C_k is solved. The domains of the variables occurring in C_k can be reduced using, e.g., consistency techniques.
2. The domain of α can be reduced as follows:

$$D_\alpha = \text{hull} \left(\bigcup_{k=1}^p (D_\alpha \cap I_k) \right).$$

A failure must happen if no I_k intersects the domain D_α .

3. If C_k is violated for some k then every element of I_k can be removed from D_α .
4. Otherwise, the constraint is satisfied in the interval sense but no domain can be reduced and the problem is still being under-constrained.

Note that the solving process must not stop before D_α takes its values in at most one I_k , otherwise the piecewise phenomenon is not taken into account and many non physics solutions may be found (case 4).

2.5 Search issues

The notions of auxiliary variables and piecewise constraints introduce several difficulties and problems:

Problem 1. The splitting steps of domains of auxiliary variables may duplicate the solutions on the main variables. For same values of the main variables, several solutions for auxiliary variables may satisfy all the constraints. This is due to some incoherent precisions between auxiliary variables and

main variables. It must also be highlighted that the set of auxiliary variables is often under-constrained, since, in the ED phase, some uncertainties remain about some product characteristics and its behavior. Thus, many solutions may be found for the same tuple of values for main variables. That may lead to useless redundant computations and to a huge number of approximate solutions corresponding to the same product architecture.

Problem 2. The main variables may not be reduced enough if the auxiliary variables are not split enough. Consistency techniques used on interval domains are based on outer approximations, which may lead to an over-estimation of variable domains. The solving process may be very long, spending most of the time in pure search on main variables, whereas auxiliary variables may have wide domains.

Problem 3. It may be difficult to choose the auxiliary variables to be split and to set precision thresholds. Proper precisions are required to efficiently manage Problem 1 and Problem 2. Moreover, some auxiliary variables are only present within the model, because they represent well-known properties of some components, phenomena, etc. However they are not required to express all the knowledge about a product. These variables and their values improve the expressivity and comprehensibility of the product model, which is important when this model may evolve as in the ED phase. Let call them functional variables, as their values are directly computed using an expression of other variables.

Problem 4. The piecewise constraints must be taken into account in order to early reduce the search space. This clearly depends on the domain of the α variables, which must be reduced to one of the I_k of the piecewise constraint to apply the constraint c_k and take into account the corresponding phenomenon.

Other issues. The ED problems are under-constrained in general. We suppose here that the precisions of main variables are well chosen enough according to the domain sizes in order to avoid a huge number of approximate solutions. Another well known approach is to specialize the search for integer variables and real variables.

3 Problem Solving

New search heuristics will be introduced to tackle the issues raised above. These heuristics will be embedded in the general interval-based branch-and-prune model.

3.1 *Branch-and-prune algorithm*

The general branch-and-prune algorithm (Van-Hentenryck et al., 1997) is defined in Algorithm 1. The input is a CSP model. The output is a set of approximate solutions enclosing the solution set.

The computation is as follows. Every domain is pruned provided that no solution (element from set 1) is lost. Every approximate solution (element from set 4) associated with the result of the proof of existence is inserted in the computed approximation. Non-empty domains are split provided that at least one of the main variables is not precise enough. The sub-problems are further solved.

The algorithm for the proof of existence validates the box computed by the *Branch-and-Prune* algorithm, and several techniques may be used according

Algorithm 1. General Branch-and-Prune Algorithm.

Solve(C : set of constraints, D : domains, (x, y) : vars) : a set of interval approximate solutions

$D := \mathbf{Prune}(C, D)$

if D is empty then

discard D

elseif D_x is precise enough then

$b := \mathbf{ProveExistence}(C, D)$

Insert(D_x, b) in the computed approximation

else

Choose a splittable variable z in (x, y)

Split($D, z, D^1 \cup D^2$)

Solve($C, D^1, (x, y)$)

Solve($C, D^2, (x, y)$)

endif

end

to the type of constraints:

- Inequality constraints can be tackled with interval computations.
- Equality constraint systems can be processed by fixed-point operators (Kearfott, 1996).

These techniques may not operate on heterogeneous and non-differentiable problems. In this case, a search process can be used to prove the existence of canonical approximate solutions, namely boxes of maximal precision satisfying the constraints in the interval sense. We consider that this smallest interval box, with closest floating-point numbers as bounds, is precise enough to claim

that we have found a solution if no inconsistencies are detected. An other approach is to apply a local search process, where the optimization function should take into account the number of inconsistent constraints balanced by the distance of violation of each one.

However, these algorithms can not always prove the existence of a solution in a box in a reasonable time. All computed solutions may not be guaranteed, but this is not the main goal for designers to have safe numerical solutions in the ED phase. All the uncertainties relating to a model make the solutions near guaranteed boxes also acceptable. However, guaranteed boxes may correspond to more robust solutions than those for which the proof of existence has failed. In the ED phase, designers are mainly interested in having an overview of the global shape of the complete space of solutions, namely, having a better insight of the feasible product architectures. When designers have an idea of some robust solutions within a solution set, they can better define the more interesting parts of this set relating to good performances criteria and robust product architectures.

3.2 Search strategies

We propose to implement several search strategies to tackle the problems described in the previous section.

Splitting ratio. The choice of variables may follow an intensification process on the main variables and a diversification strategy on the auxiliary variables. The idea is to limit the duplication of solutions (Problem 1) and to compute efficient reductions on the whole system (Problem 2). A diversification process

aims at gathering some knowledge on the problem, whereas an intensification process uses this knowledge to explore and to focus on interesting areas of the search space (Blum & Roli, 2003). The intensification/diversification strategy can be controlled by a ratio between the two types of variables to choose (see Algorithm 2). Inside each group, a round robin strategy may be used to make the algorithm robust. A high ratio corresponds to high intensification on main variables and a small one increases diversification on auxiliary variables.

Algorithm 2. Search heuristic favoring main variables

SelectVariable(X : set of variables, D : domains, R : integer ratio)

$X_m := \{x \in X : x \text{ is a main var.}, D_x \text{ can be split}\}$

$X_a := \{x \in X : D_x \text{ can be split}\} \setminus X_m$

let n be the number of carried out splits

let n_m be the number of splits on main var.

if X_a is empty **or** $n = 0$ **or** $n_m < R(n - n_m)$

$n_m := n_m + 1$

$x := \text{SelectRoundRobin}(X_m)$

else

$x := \text{SelectRoundRobin}(X_a)$

endif

$n := n + 1$

return x

end

This heuristic is applicable to any ED problem, since ED problems always include some main variables (which values statements are the main objective of the ED phase). Moreover, these variables are often useful to compute rel-

evant values for auxiliary variables, since auxiliary variables have to express some characteristics (physics phenomenon, geometry, etc.) of a specific product architecture. Main variables are better defined (small domains and accurate precisions according to the product specifications) than auxiliary variables (for instance: complex phenomena with several simplifying assumptions). In this way, the constraint propagation phase may be more interesting in reducing domains of auxiliary variables than the splitting steps on this huge search space.

Precision. Two types of auxiliary variables can be identified (Problem 3). Auxiliary variables expressed as functions of other variables may not be split since they correspond to intermediate computations. To this end, it suffices to bind these variables to an infinite precision. Their values are computed using the Prune algorithm (constraint propagation). The other auxiliary variables may be split (Problem 2), but their precisions have to be as relevant as possible to avoid too many useless splitting steps (Problem 1).

Piecewise constraint. The goal is to split the α variable according to the first pruning case of the constraint in order to answer Problem 4. The domain of α must be included in some I_k in order to enforce C_k . To this end the domain of α can be split on the bounds of the intervals I_k instead of the classical bisection. Let us note that even the auxiliary variables with infinite precision must be considered here. Combining several piecewise constraints parametrized by the same variable boils down to considering the set of bounds from all the intervals I_k and to combine the constraints from the corresponding pieces.

Variable types. A common approach is to choose first integer variables and then real variables, supposing that different integer values may correspond

to different product architectures. We then have several choice criteria to be combined: type of variable (main, auxiliary), domain nature (discrete, continuous), and more usual criteria (round-robin strategy, largest continuous domain, smallest discrete domain, most constrained variable, etc.). Integer variables are supposed to be enumerated and real variables are bisected.

3.3 Representation

Several approximate solutions are redundant if the domains of the main variables intersect, because of the search on auxiliary variables. In this case, they need to be merged in order to compute compact representations of the solution set. In the interval framework a set of merged boxes can be replaced by their hull, namely the smallest box containing each element.

It must be verified that the main variables are still precise enough after merging. In particular, several boxes enclosing a continuum of solutions may share only some bounds. The hull may not be computed to keep fine-grained approximations.

4 Empirical evaluation on academical problems

The techniques have been implemented in Realpaver (Granvilliers & Benhamou, 2006). The pruning step is implemented by constraint propagation using 2B consistency and box consistency. The next results do not take into account the computations of any proof of existence algorithm, since only performances of search heuristics are studied. These results are only concerned of finding solutions which are coherent with precisions of variables. The presented

curves show the number of splits made on domains of variables. Considering one solving heuristic, this number does not vary on the contrary to the solving time, which depends on the computer hardware, the operating system, other running process, etc. Moreover, it unmistakably represents the performances of each search heuristics, since we do not interfere with the pruning algorithm.

4.1 Functional variables

ED problems embody many variables expressed as functions of other variables. They are maintained within the model to preserve the model intelligibility, although they could be removed and replaced by their expression. The question is whether these variables have to be split. Let us consider the following problem parametrized by $n \geq 3$:

$$\left\{ \begin{array}{l} x_k \in [-100, 100] \quad 1 \leq k \leq n \\ y_k = x_k^2 - x_{k+1}^2 \quad 1 \leq k \leq n \\ y_k - y_{k+1} = k \quad 1 \leq k \leq n \end{array} \right. \quad (8)$$

Let x_{n+1} be x_1 and let y_{n+1} be y_1 . The goal is to prove that the problem has no solution. The results are depicted in Figure 3. The \bullet curve corresponds to a round robin strategy on x and no split on y . This is clearly not efficient. The \blacksquare curve is obtained with a round robin strategy on x and y . The growth ratio is almost the same (factor 2) but the number of splitting steps is decreased by a factor 50. The \blacktriangle curve is derived by a more robust strategy such that x is split twice more than y . Surprisingly the number of splitting steps decreases when n increases. For instance, given $n = 8$, the number of bisections is respectively

93183, 1791, and 93 for the three heuristics.

Let us consider another problem parametrized by $n \geq 3$:

$$\left\{ \begin{array}{ll} x_k \in [-\pi/3, \pi/3] & 1 \leq k \leq n \\ y_k = x_{k+1} + x_{k+2} & 1 \leq k \leq n \\ \tan(x_k + y_k) + \tan(x_k) = k/n & 1 \leq k \leq n \end{array} \right. \quad (9)$$

such that $x_{n+i} = x_i$ and $y_{n+i} = y_i$ for every $i \geq 1$. The goal is to compute the solutions on x considering a precision of 10^{-8} (three solutions for $6 \leq n \leq 11$). The results are depicted in Figure 4. The \bullet curve corresponds to a round robin strategy on x and no split on y . The \blacksquare curve is obtained with a round robin strategy on x and y . We see that it is more efficient not to split y . The other curves are obtained with a robust strategy such that x is split r times more than y ($r = 5$ for \blacktriangle and $r = 10$ for \blacklozenge). The improvement increases with ratio r .

The previous results may lead to the following conclusions. In the first problem, every reduction on functional variables is directly propagated through many constraints, which is efficient since these variables occur in several constraints. If such variables appear in only one constraint, splitting them is not efficient, because they only represent intermediary computations. The second problem shows that no split on functional variables gives bad performances. In fact, every reduction on y_k leads immediately to a reduction of x_{k+1} and x_{k+2} since the constraint is simple. This is a means for tackling two variables using only one split. Finally, strategies using a ratio are more robust and efficient than others on these types of problems.

It can be noted that in ED models, functional variables often take part of under-constrained network of constraints. Many splitting steps on them is useless and a high ratio is better. If this ratio is too difficult to establish, no splitting steps on functional variables is the easiest and the more efficient approach. Moreover all splitting steps on functional variables do not have the same impact on the pruning of the whole problem and the round robin strategy does not take this factor into account. Perhaps, some other strategies, as for instance to choose the most constrained variable, should be more efficient especially with small ratios, where functional variables are often split.

4.2 Auxiliary variables

Auxiliary variables are useless from an ED point of view but they have to be efficiently managed during computation. Let us consider the following problem where n is an integer main variable in $[-10^8, 10^8]$, x, y, z are real variables in $[-10, 10]$ with precision 10^{-8} , x is a main variable, y and z are auxiliary variables:

$$\left\{ \begin{array}{l} x - y + z = 1 - n \\ x - yz = 0 \\ x^2 - y + z^2 = 2 \end{array} \right. \quad (10)$$

The problem projected onto the auxiliary variables is hard to solve, since this problem is dense. Local reasoning about projections may not compute efficiently domains of variables. As a consequence these variables must often be split. The curve in Figure 5 is obtained from a robust strategy that al-

where n is the number of pieces of the piecewise constraint, each interval I_k is defined by

$$I_k = \left[-10 + 20\frac{k-1}{n} + \nu, -10 + 20\frac{k}{n} - \nu \right], 1 \leq k \leq n, \quad (12)$$

$\nu > 0$ is equal to the machine precision, and $\text{mid}(I_k)$ is the midpoint of I_k . Let 10^{-8} be the precision of every variable.

Figure 6 depicts the number of splitting steps required for solving the problem parametrized by the number of pieces of the piecewise constraint. The variables are chosen following a round robin strategy. The \blacklozenge curve is such that only the first pruning case of the piecewise constraint is applied. A restricted pruning algorithm is clearly not efficient. In this case, many approximate solutions include piece bounds and the piecewise constraint is useless. The \bullet curve corresponds to a full pruning algorithm with classical bisection, the \blacksquare one to a full pruning algorithm with split on the first hole from the domain of x , and the \blacktriangle one to a full pruning algorithm with split on the mid hole. Bisection is more efficient than split on the first hole. It is known that bisection is more efficient than labeling for continuous variables. Split on the mid hole is the best heuristics. The corresponding function follows n^p with $0 < p < 1$. The new technique seems effective even for huge piecewise constraints.

5 Empirical evaluation on real-world problems

In this section we evaluate our approach on models obtained for real world applications in mechanical engineering. It may be noted that the next results do not take into account the existence proof algorithm, because of the huge

number of computation steps it adds. Moreover we are mainly interested in studying our search heuristic results.

5.1 *A basic batch-exchanger system*

We first consider a batch exchanger model (see Figure 7). The solution principles are defined using five design variables (three catalogs related to lengths, materials and diameters, the number of fins, and the gap between fins). In this model, the variables relating to the choices within catalogs are design variables instead of the length and materials of fins and the diameter of the tube, which ones have their values directly defined by the catalog. This problem is interesting from an ED point of view, since we have to choose several components in a (small) catalogs, while dimensioning the gap between fins. In this system, there is a coupling between fluid mechanics and the geometry of the heat exchanger (namely the gap between fins). System modeling introduces five auxiliary variables and five functional variables. The batch-exchanger is part of a batch-cooler system for aperitif and this model investigates the feasibility to cool down the aperitif from 25°C to 8°C in less than 25 seconds.

Figure 8 depicts the number of splitting steps using the robust strategy with ratio r . The number of solutions is half of the number of splitting steps. The ■ curve comes from labeling of integer domains and the ▲ curve from bisections. We see that a high splitting ratio allows to decrease the number of splitting steps. Due to some orientation in the model the auxiliary variables are directly computed from the design variable values. Splitting auxiliary variables leads to duplicate solutions and consequently useless search steps. Bisection on integer variables is better than labeling. That is explained by efficient reductions of

the number of fins (integer in $[5, 20]$) using some bound consistency method.

5.2 *A pump and tank water circuit*

This model takes into account three tanks (one upstream and two downstream) and one water pump (see Figure 9). The objective is to study the feasibility of dimensioning the two lines diameters after the downstream Y-branch. Before the Y-branch, the lines diameter is 0.055 meter. All the lines lengths are fixed and the two downstream tanks must receive the same flow, considering that the sum of the global lines section is the same before and after the Y-branch. The water pressures in the tanks are defined initially: the upstream tank is at 40000 pascal and the two downstream tanks are open to the atmosphere air and the pressure is 101325 pascal. The pump is standardized and has characteristics (efficiency, manometric head and required net positive suction head for a water flow, etc.) given by its manufacturer. The net positive suction head is investigated to guarantee the safety of the pump. The solutions are computed taking into account that the cavitation phenomenon in the pump must not appear, otherwise it may be seriously damaged. The downstream circuit (directly linked to the cavitation phenomenon) is coupled to the whole circuit (pressure losses) and the Y-branch make the problem non trivial.

This model is made of two design variables (the two tube diameters after the Y-branch), three auxiliary variables and thirty-five functional variables. The figures 10, 11, 12, 13, 14 and 15 depict the results obtained when functional variables are split considering a global varying precision. Since the three auxiliary variables have a fixed precision, a global precision can be defined on the other variables, i.e. functional variables and they are split like auxiliary

variables. Half of those pictures depicts the numbers of splits and the other the numbers of solutions. The \bullet curves show the results obtained with a classical round-robin strategy for the choices on all variables (main, auxiliary and functional variables). The \blacksquare curves express the results with a strategy always starting with main variables. Once they reach the required precision, auxiliary are variables split. The \blacktriangle curves represent the results with a choice strategy with a ratio defining a priority of 3 for the main variables on the auxiliary variables. Only results found within a reasonable time are written out on each curve: results with a solving time exceeding one hour are not taken into account.

The figures 10 and 11 represent the most general case and all the functional variables are defined with the same global precision. The different number of solution between each run can be explained by the miscellany of the duplication of design solutions and the powerlessness of consistency algorithm on intervals, which never remove real solutions, but have difficulty to prune accurately some domains and to reject them if they are near a real solution. In this context, the most accurate precision on functional variables given reasonable solving time is 10^{-1} . The \bullet curve seems to have the worst results, in particular for the more accurate precision, but otherwise the results are fairly similar. It may be noted that merging all the computed solutions gives only one design architecture. Considering that fact, the best computing run is obtained by the \blacktriangle curve with 3 solutions and 296 splits for a functional variables precision of 10^3 . The best approach considering the whole curves seems to be the \blacktriangle curve, where a robust strategy is applied.

After these first results, we can observe that the precision 10^3 and 10^4 for functional variables give good results. These quantities are compatible with some

functional variables values: losses in lines expressed in pascal and Reynolds number values. So the figures 12 and 13 give results where these functional variables have several fixed precisions. In this case, the most accurate precision is 10^{-10} . Globally the \bullet curve seems to be again the worst approach, although it gives the smallest number of solutions after a precision of 10^{-7} for the same quantity of splits than others. The lowest number of solutions is given by the \blacksquare with 3 solutions for 148 splits for a precision of 10^0 . Previously the same small number of solutions was found, but in 296 splits.

From the maximum precision to 10^1 the results stay the same, but until 10^{-2} the number of solutions and the number of splits decrease. We can conclude that some other functional variables values are compatible with these precisions. Then set precisions are also given for the net positive suction head and the total manometric head and all surfaces. The figures 14 and 15 show the results obtained with all these set precisions and with only a few still using the varying global precision. In this case, the robust strategy fails to give all the solutions within reasonable time after a precision of 10^{-8} , although it seems to give the best results before a global precision of 10^{-3} . The two other curves allow a maximum precision of 10^{-11} and up to a precision of 10^{-7} the results are interesting. The best run is obtained by the \blacksquare curve with 3 solutions and 148 splits for precisions of 10^0 and 10^{-1} , which is not better than in the previous case.

With these results, we can conclude that functional variables have to be split (using CSP based on interval arithmetic). But it is difficult to define carefully the precision on each functional variable. If the quantities represented by their values are known by designers, well defined precisions can be set, but otherwise few splits are better not to duplicate main variables solutions.

5.3 *A bootstrap problem*

A basic model of an aircraft conditioning system is investigated (see Figure 16). Air coming from a turbo-reactor and from the atmosphere is used to produce cold air. The atmosphere air cools down the air coming from the turbo-reactor through a heat exchanger where complex piecewise defined physics phenomena are studied (Fanning friction factor and Nusselt number). Turbo-reactor air flow passes through a compressor to improve the heat transfer phenomenon inside the exchanger. Before exiting the air conditioning system, a turbine releases its pressure and makes its temperature decrease. A coupling shaft conveys the turbine mechanical energy to the compressor. This problem is difficult to solve since many physics phenomena interfere. The loop corresponding to the bootstrap make its components coupled according to the temperatures and pressures of the air flux. These temperatures and pressures are also linked to the heat exchanger geometry (gap between plates).

In this model, the compressor, the turbine and the coupling shaft are standardized components and only the heat exchanger has to be embodied as it mainly determines the air-conditioning performances. The main objective of the system is to bring air to the passengers and the crew of the aircraft and to control the air temperature and pressure inside the cockpit. But some other criteria are important in an aircraft, as the air flow taken from the turbo-reactor (that decreases its efficiency), the increase of the aircraft drag, the weight of the air-conditioning system, etc. This problem is efficiently solved with piecewise constraints: 6734 splitting steps and 1262 approximate solutions with respect to 36978 splitting steps and 18860 approximate solutions without piecewise constraints.

Moreover this problem cannot be solved within reasonable time with classical round-robin strategies on all the variables. The search space is so wide, that if the embodiment design knowledge about main variables is not used, the solving process becomes very long. The use of functional variables with infinite precision is the easiest way, since the model is complex and functional variables values quantities can change. For instance, the Reynolds number takes its values from 100 to 200000.

It may be noted in the solution set of these real problems, that auxiliary variables precision are often large. Indeed the interval approach may compute interval solutions, where each one may contain several solutions over the real numbers. If the model is very sensitive to main variables values and if their precisions are not small enough, auxiliary variables may have large domains since they correspond to the several main variables real values, which are contained in one interval. From the designers' point of view and in the context of ED, it does not matter, because the main goal is to investigate the feasibility of design concepts. Designers' first interest is to know where there is no solution in the search space. If they want to have more precise auxiliary variables values for one specific design architecture, they just have to change all variables domains corresponding to one or several computing solution values and then to increase main variables precision. They can start a new solving step on this more restricted search space and find more accurate values.

6 Related work

Constraint techniques may be used at two successive stages of preliminary design. Discrete constraints may lead to determine the architecture of a product

during the conceptual design phase (O’Sullivan, 2001). CD using components from the shelf is known as configuration. These problems can be represented by dynamic constraint satisfaction problems (Mittal & Falkenhainer, 1990) such that the involved components are activated and the corresponding constraints are solved. The notion of component (or variable) activation can be tackled by conditional constraints (Gelle & Faltings, 2003; Sabin et al., 2003). Larger and more complex problems are also tackled by (Stumptner et al., 1998; Mailharro, 1998). From a solving point of view the main goal is to efficiently traverse the tree of architectures. Numerical nonlinear constraints are more involved in the ED phase. The frontier between CD and ED may be thin because mixed constraints can be considered (Gelle & Faltings, 2003) to tackle both phases at the same time. But the ED physics models are in general more complex.

Sam-Haroud & Faltings (1996) have proposed to represent numerical constraints by 2^k -trees, namely decompositions of the feasible regions using interval boxes. Strong consistency techniques have been defined through the combination of 2^k -trees. Design applications such as bridge design have been efficiently solved. In this framework, constraint systems are decomposed in binary and ternary constraints in order to limit the size of 2^k -trees (quadtrees if $k = 2$ and octrees if $k = 3$).

Classical interval techniques have been implemented in the ED platform Constraint Explorer (Zimmer & Zablit, 2001). The solving engine combines interval arithmetic, constraint propagation and search. An important feature is the analysis of the constraint network using graph decomposition (Bliek et al., 1998). The result of this analysis is an ordering of variables to be fixed before solving the associated constraint blocks. Recent developments can be found in (Neveu et al., 2006). Our approach can be directly integrated for solving

one block. In particular large blocks may arise in ED models, for instance in the study of the equilibrium of a system.

Piecewise constraints can be implemented by means of conditional constraints (Zimmer & Zablit, 2001). This method amounts to the first case of our pruning algorithm. More recently, binary piecewise constraints with pieces in the form of $(x, y) \in I_k \times J_k : C_k(x, y)$ have been represented by quadtrees (Vareilles et al., 2005). It seems difficult to extend this approach to constraints of higher arities, which is required for solving the problems described in this paper.

Solution sets with nonzero volumes may be characterized by inner approximations, namely interval boxes of which every point is a solution. Several works have tackled specific cases: inequality constraints by means of 2^k -trees (Sam-Haroud & Faltings, 1996), interval boxes (Collavizza et al., 1999) or the extreme vertex representation (Vu et al., 2005), and equality constraint systems with at least as many existential quantifiers as equations (Goldsztein & Jaulin, 2006). The study of such techniques for more heterogeneous constraint systems is an issue.

Other works have taken into account relations that are not described by analytical expressions (Yannou et al., 2003; Fischer et al., 2004), exploiting in the constraint framework simulation results or data from black box numerical tools. The main idea is to compute approximate constraint-based models of these relations.

7 Conclusion

ED problems have been represented by constraint satisfaction problems with existential quantifiers. ED knowledge on types of variables and precisions has been used to improve the solver efficiency. New search heuristics based on a splitting ratio have been introduced to tackle the quantified variables. Duplicated solutions of main variables disappear and decisions on the design solution principles set are easier to make for designers. A global constraint has been defined for piecewise defined physics phenomena. Experimental results from academic and real-world problems are promising. Embodiment design goals are better taken into account since the main purpose is to investigate the feasibility of the search space.

There are many directions for future research. The notion of splitting ratio could be refined to tackle the hardness of every variable. The hardness of a variable should be clearly defined. For instance, dependencies between variables may also indicate variables relevancy in the model and possibly participate to their hardness. Auxiliary variables precision and solutions validation could be more studied. The notion of precision is essential in numerical computations. The precision on auxiliary variables is not often chosen appropriately and it induced many useless computations steps in all heuristic search. The precision on main variables is easily defined considering the design knowledge about the model: epistemic knowledge about main variables values. On the other hand, auxiliary variables are often part of complex mathematical expression. In fact, the sensitivity of each variable should be investigated and precision should be defined considering the numerical analysis of each constraint in which variables are involved. Nevertheless in practice, it is very difficult to apply and designers

have no time to investigate in those fastidious calculations. Moreover, the integration of our techniques in a block solving approach could be explored. The block decomposition of a CSP takes into account the constraints network and established an order or a causality on variables or blocks of variables based on this network. In most design models, starting variables are needed to compute a relevant order, since models are often under-constrained. Several orders on variables may be defined for the same constraint graph, and the choice of the optimal one is undecidable within reasonable time (Jégou & Terrioux, 2003), but the main variables heuristic may help in this ordering task, taking into account design knowledge.

References

- Beame, P., Cook, S., Edmonds, J., Impagliazzo, R., and Pitassi, T. (1995). The relative complexity of NP search problems. *In Proceedings of the Twenty-Seventh Annual ACM Symposium on theory of Computing* (Las Vegas, Nevada, United States, May 29 - June 01, 1995). STOC '95. ACM Press, New York, NY, 303-314.
- Blum, C., & Roli, A. (2003), Metaheuristics in combinatorial optimization : Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268-308.
- Benhamou, F., Goualard, F., & Granvilliers, L., & Puget, J.-F. (1999). Revising Hull and Box Consistency. In *International Conference on Logic Programming*, pages 230-244. The MIT Press.
- Bliet, C., & Neveu, B., & Trombettoni, G. (1998). Using Graph Decomposition for Solving Continuous CSPs. In *CP'98*, Pisa, Italy.
- Collavizza, H., & Delobel, F., & Rueher, M. (1999). Extending Consistent

- Domains of Numeric CSPs. In *IJCAI'99*, Stockholm, Sweden.
- Cook, S., A., & Mitchell, D., G. (1997). Finding Hard Instances of the Satisfiability Problem: A Survey. DIMACS Series in Discrete Math. and Theoretical Computer Science, 35, 1997, pp1-17.
- Fischer, X., & Sébastien, P., & Nadeau, J.-P., & Zimmer, L. (2004). Constraint based Approach Combined with Metamodeling Techniques to Support Embodiment Design. In *SCI'04*, Orlando, USA.
- Gelle, E., & Faltings, B. (2003). Solving Mixed and Conditional Constraint Satisfaction Problems. *Constraints*, 8:107-141.
- Goldsztejn, A., & Jaulin, L. (2006). Inner and Outer Approximations of Existentially Quantified Equality Constraints. In *CP'06*, Nantes, France.
- Granvilliers, L., & Benhamou, F. (2006). Algorithm 852: Realpaver: An interval solver using constraint satisfaction techniques. *ACM TOMS*, 32(1):138-156.
- Hyvönen, E. (1989). Constraint Reasoning Based on Interval Arithmetic. In *IJCAI'89*, Detroit, USA.
- Jégou, P., Terrioux, C. (2003). Hybrid backtracking bounded by tree-decomposition of constraint networks. *Artificial Intelligence*, vol. 146, pp. 43-75.
- Kearfott, R. B. (1996). *Rigorous Global Search: Continuous Problems*. Non-convex Optimization and Its Applications. Kluwer Academic Publishers.
- Lhomme, O. (1993). Consistency Techniques for Numeric CSPs. In *IJCAI'93*, Chambéry, France.
- Mailharro, D. (1998). A classification and constraint-based framework for configuration. *AI EDAM*. 12(4). 383-397.
- Mittal, S. & Falkenhainer, B. (1990). Dynamic Constraint Satisfaction Problems. In *AAAI'90*, Boston, USA.

- Moore, R. (1966). Interval Analysis. Prentice-Hall.
- Neveu, B., & Chabert, G., & Trombettoni, G. (2006). When Interval Analysis Helps Interblock Backtracking. In *CP'06*, Nantes, France.
- O'Sullivan, B. (2001). Constraint-Aided Conceptual Design. Professional Engineering Publishing.
- Pahl, G., & Beitz, W. (1996). Engineering Design: A Systematic Approach. Springer.
- Richardson, D. (1968). Some Unsolvable Problems Involving Elementary Functions of a Real Variable. *Journal of Symbolic Logic*, 33, 514–520.
- Rossi, F., & van Beek, P., & Walsh, T. (2006). Handbook of Constraint Programming. Elsevier.
- Rothwell, R., & Gardiner, P. (1990). Robustness and Product Design Families, *Design Management: A Handbook of Issues and Methods*, pp. 279-292. (Oakley, M., ed.), Basil Blackwell Inc., Cambridge, MA
- Sabin, M., & Freuder, E. C., & Wallace., R. J. (2003). Greater Efficiency of Conditional Constraint Satisfaction. In *CP'03*, Kinsale, Ireland.
- Sam-Haroud, D., & Faltings, B. (1996). Consistency Techniques for Continuous Constraints. *Constraints*, 1:85-118.
- Sébastien, P., & Chenouard, R., & Nadeau, J.-P., & Fischer, X.(2006). The Embodiment Design Constraint Satisfaction Problem of the BOOTSTRAP facing interval analysis and Genetic Algorithm based decision support tools, Proceedings of Virtual Concept 2006, Mexico.
- Stumptner, M., & Friedrich, G., & Haselböck, A. (1998). Generative constraint-based configuration of large technical systems. *AI EDAM* 12(4), 307-320.
- Van-Hentenryck, P., & Mc Allester, D., & Kapur, D. (1997). Solving Polynomial Systems Using Branch and Prune Approach, *SIAM Journal on Nu-*

- merical Analysis*, vol. 34(2), p. 797-827.
- Vareilles, E., & Aldanondo, M., & Gaborit, P., & Hadj-Hamou, K. (2005). Using Interval Analysis to Generate Quadrees of Piecewise Constraints. In *IntCP'05*, Barcelona, Spain.
- Vu, X., & Sam-Haroud, D., & Silaghi, M. (2002). Approximation Techniques for Nonlinear Problems with Continuum of Solutions. In *SARA'02*, Kananaskis, Canada.
- Wang, P. S. (1974). The Undecidability of the Existence of Zeros of Real Elementary Functions. *J. ACM* 21, 4 (Oct. 1974), 586-589.
- Yannou, B., & Simpson, T. W., & Barton, R. R. (2003). Towards a Conceptual Design Explorer using Metamodeling Approaches and Constraint Programming. In *ASME DETC'03*, Chicago, USA.
- Zimmer, L. & Zablitz, P. (2001). Global Aircraft Predesign based on Constraint Propagation and Interval Analysis. In *CEAS-MADO'01*, Koln, Germany.

A Batch-exchanger model

Constants names and values:

The batch volume (cl):	$V := 6$
Fin thickness (mm):	$e_{ail} := 0.5$
Initial temperature of the aperitif ($^{\circ}C$):	$T_i := 20$
Final temperature of the aperitif ($^{\circ}C$):	$T_f := 8$
Volume of aperitif to cool down (cl):	$dose := 4$

Design Variables names, domains and precisions:

Catalog for the fins materials (-):	$mater \in \{1, 2\}$
Catalog for the fins length (-):	$ail \in \{1, 2\}$
Catalog for the tube diameter (-):	$diam \in \{1, 2\}$
Number of fins (-):	$N \in [5..20]: \text{integer}$
Space between fins (mm):	$e \in [1..4]: p(e) = 10^{-1}$

Auxiliary variables names and domains:

Time to cool down the aperitif (s):	$t \in [11..15]: p(t) = 10^{-1}$
Tube diameter (mm):	$d \in [0..50]: \text{Integer}$
fins length (mm):	$L \in [0..50]: \text{Integer}$
Fin conductivity (W/m/K):	$\lambda \in [1..200]: \text{Integer}$
Saturation temperature ($^{\circ}C$):	$T_{sat} \in [-15..2]: p(T_{sat}) = 10^{-1}$

Functional Variables names and definition:

Surface of a semi fin (m^2):	$A_{ail} = \frac{L^2 - \frac{\pi}{4} \cdot d^2}{1000000}$
----------------------------------	---

Exchanger surface (m ²):	$A = \frac{N \cdot (2 \cdot A_{ail} + \frac{\pi \cdot e \cdot d}{1000000}) \cdot dose}{V}$
Exchange coefficient int the batch-exchanger (-):	$h = \frac{1200}{e}$
Efficiency coefficient for a fin (-):	$f_i = \frac{L-d}{2000} \cdot \sqrt{\frac{2000 \cdot h}{\lambda e_{ail}}}$
Fin efficiency (-):	$\eta = \frac{e^{(2 \cdot f_i)} - 1}{e^{(2 \cdot f_i)} + 1}$

Constraints:

Balance of heat energy	$T_f = T_{sat} + (T_i - T_{sat}) \cdot e^{\frac{-h \cdot A \cdot t \cdot \eta}{39 \cdot dose}}$
Batch volume	$V = e \cdot A_{ail} \cdot N \cdot 100$
Catalog of tube diameters	$diam = 1 \rightarrow d = 16$ $diam = 2 \rightarrow d = 18$
Catalog of fin materials	$mater = 1 \rightarrow \lambda = 200$ $mater = 2 \rightarrow \lambda = 20$
Catalog of fin length	$ail = 1 \rightarrow L = 40$ $ail = 2 \rightarrow L = 50$

B Pump and tank water circuit model

Constants names and values:

Pressure in the upstream tank (pa):	$P_{amont} := 40000$
Pressure in the downstream tanks (pa):	$P_{aval} := 101325$
Height of the vertical downstream line before the Y-branch (m):	$H_{r1} := 5$
Height of the vertical downstream line after the Y-branch (m):	$H_{r2} := 2$
Height of the vertical upstream line (m):	$H_a := 2$
Height of water in the upstream tank (m):	$H_w := 0.5$
Water density (kg/m ³):	$\rho := 1e3$

Water viscosity (m ² /s):	$\mu := 1e - 3$
Acceleration due to gravity (m/s ²)	$g := 9.81$
Lines diameter before the Y-branch(m):	$D := 0.055$
Losses coefficient in entry of upstream line:	$\xi_1 := 0.5$
Losses coefficient exiting downstream lines:	$\xi_3 := 1$
Losses coefficient in the Y-branch towards the first downstream tank:	$\xi_4 := 0.5$
Losses coefficient in the Y-branch towards the second downstream tank:	$\xi_5 := 0.1$
Water temperature (°C):	$T := 13$

Design Variables names, domains and precisions:

Line diameter after the Y-branch towards the first downstream tank (m):	$D_{r1} \in [0.02, 0.1]: p(D_{r1}) = 10^{-3}$
Line diameter after the Y-branch towards the second downstream tank (m):	$D_{r2} \in [0.03, 0.1]: p(D_{r2}) = 10^{-3}$

Auxiliary variables names and domains:

Flow in the lines before the Y-branch:	$Q_0 \in [17/3600, 96/3600]: p(Q_0) = 10^{-5}$
Flow in the lines after the Y-branch towards the first downstream tank:	$Q_{r1} \in [0, 96/3600]: p(Q_{r1}) = 10^{-5}$
Flow in the lines after the Y-branch towards the second downstream tank:	$Q_{r2} \in [0, 96/3600]: p(Q_{r2}) = 10^{-5}$

Functional Variables names and definition:

Section of cylindrical upstream lines	$S = \frac{\pi \cdot D^2}{4}$
---------------------------------------	-------------------------------

(m ²):	
Section of cylindrical downstream lines towards the first tank (m ²):	$S_{r1} = \frac{\pi \cdot D_{r1}^2}{4}$
Section of cylindrical downstream lines towards the second tank (m ²):	$S_{r2} = \frac{\pi \cdot D_{r2}^2}{4}$
Surface of the vertical upstream line (m ²):	$A_{e1} = \pi \cdot D \cdot H_a$
Surface of the horizontal upstream line (m ²):	$A_{e2} = \pi \cdot D \cdot L_a$
Surface of the vertical downstream line before the Y-branch (m ²):	$A_{e3} = \pi \cdot D \cdot H_{r1}$
Surface of the horizontal line towards the first downstream tank (m ²):	$A_{e4} = \pi \cdot D_{r1} \cdot L_{r1}$
Surface of the vertical downstream line towards the second tank (m ²):	$A_{e5} = \pi \cdot D_{r2} \cdot H_{r2}$
Surface of the horizontal line towards the second downstream tank (m ²):	$A_{e6} = \pi \cdot D_{r2} \cdot L_2$
Flowing speed in the lines before the Y-branch (m/s):	$V_0 = \frac{Q_0}{S}$
Reynolds number for the water before the Y-branch (-):	$Re_1 = \frac{\rho \cdot V_0 \cdot D}{\mu}$
Piecewise definition of Fanning friction factor for flowing before the Y-branch:	$Re_1 \in [0, 2100] \rightarrow f_1 = \frac{16}{Re_1}$ $Re_1 \in [2100, 50000] \rightarrow f_1 = 0.10512 \cdot Re_1^{-0.244}$ $Re_1 \in [50000, 1000000] \rightarrow f_1 = 0.04234 \cdot Re_1^{-0.164}$
Reynolds number for the water between the Y-branch and the first downstream tank(-):	$Re_2 = \frac{\rho \cdot \frac{Q_{r1}}{S_{r1}} \cdot D_{r1}}{\mu}$
Definition of Fanning friction factor for	$Re_2 \in [0, 2100] \rightarrow f_2 = \frac{16}{Re_2}$

flowing between the Y-branch and the tank 1:	$Re_2 \in [2100, 50000] \rightarrow f_2 = 0.10512 \cdot Re_2^{-0.244}$ $Re_2 \in [50000, 1000000] \rightarrow f_2 = 0.04234 \cdot Re_2^{-0.164}$
Reynolds number for the water between the Y-branch and the second downstream tank (-):	$Re_3 = \frac{\rho \cdot \frac{Q_{r2}}{S_{r2}} \cdot D_{r2}}{\mu}$
Definition of Fanning friction factor for flowing between the Y-branch and the tank 1:	$Re_3 \in [0, 2100] \rightarrow f_3 = \frac{16}{Re_3}$ $Re_3 \in [2100, 50000] \rightarrow f_3 = 0.10512 \cdot Re_3^{-0.244}$ $Re_3 \in [50000, 1000000] \rightarrow f_3 = 0.04234 \cdot Re_3^{-0.164}$
Losses coefficient in the upstream elbow (pa):	$\xi_2 = 0.15 + 0.0175 \cdot 4 \cdot f_1 \cdot 2 \cdot 90$
Losses coefficient in the downstream elbow (pa):	$\xi_6 = 0.15 + 0.0175 \cdot 4 \cdot f_3 \cdot 2 \cdot 90$
Total manometric head (m):	$H = -1.1763 \cdot 10^{-5} \cdot (Q_0 \cdot 3600)^3 - 2.2052 \cdot 10^{-4} \cdot (Q_0 \cdot 3600)^2 + 1.4384 \cdot 10^{-2} \cdot (Q_0 \cdot 3600) + 21.554$
Net positive suction head required:	$NPSH_r = 1.2144 \cdot 10^{-5} \cdot (Q_0 \cdot 3600)^3 - 1.2301 \cdot 10^{-3} \cdot (Q_0 \cdot 3600)^2 + 4.9136 \cdot 10^{-2} \cdot (Q_0 \cdot 3600) + 0.49957$
Net positive suction head available:	$NPSH_a = \frac{P_{amont} - P_{sat}}{\rho \cdot g} + (H_a + 2 \cdot D) - \frac{DP_0 + DP_1 + DP_2 + DP_3}{Ro \cdot g}$
Water saturation vapour pressure (pa):	$P_{sat} = e^{23.3265 - \frac{3802.7}{T + 273.18} - (\frac{472.68}{T + 273.18})^2}$
Total losses in the circuit (pa):	$\Delta P = \Delta P_0 + \Delta P_1 + \Delta P_2 + \Delta P_3 + \Delta P_4 + \Delta P_5 + \Delta P_6 + \Delta P_7$
Losses in entry of the vertical upstream line (pa):	$\Delta P_0 = \frac{\xi_1 \cdot \rho \cdot V_0^2}{2}$

Losses in the vertical upstream line (pa):	$\Delta P_1 = \frac{f_1 \cdot A_{e1}}{S^3} \cdot \frac{\rho \cdot Q_0^2}{2}$
Losses in the upstream elbow (pa):	$\Delta P_2 = \frac{\xi_2 \cdot \rho \cdot V_0^2}{2}$
Losses in the horizontal upstream line (pa):	$\Delta P_3 = \frac{f_1 \cdot A_{e2}}{S^3} \cdot \frac{\rho \cdot Q_0^2}{2}$
Losses in the vertical downstream line before the Y-branch (pa):	$\Delta P_4 = \frac{f_1 \cdot A_{e3}}{S_{r1}^3} \cdot \frac{\rho \cdot Q_0^2}{2}$
Losses in the Y-branch towards the first downstream tank (pa):	$\Delta P_5 = \frac{\xi_4 \cdot \rho \cdot \left(\frac{Q_{r1}}{S}\right)^2}{2}$
Losses in the horizontal line towards the first downstream tank (pa):	$\Delta P_6 = \frac{f_2 \cdot A_{e4}}{S_{r1}^3} \cdot \frac{\rho \cdot Q_{r1}^2}{2}$
Losses exiting the line in the first downstream tank (pa):	$\Delta P_7 = \frac{\xi_3 \cdot \rho \cdot \left(\frac{Q_{r1}}{S_{r1}}\right)^2}{2}$
Losses in the Y-branch towards the second downstream tank (pa):	$\Delta P_8 = \frac{\xi_5 \cdot \rho \cdot \left(\frac{Q_{r2}}{S_{r2}}\right)^2}{2}$
Losses in the vertical downstream line after the Y-branch (pa):	$\Delta P_9 = \frac{f_3 \cdot A_{e5}}{S_{r2}^3} \cdot \frac{\rho \cdot Q_{r2}^2}{2}$
Losses in the elbow towards the second downstream tank (pa):	$\Delta P_{10} = \frac{\xi_6 \cdot \rho \cdot \left(\frac{Q_{r2}}{S_{r2}}\right)^2}{2}$
Losses in the horizontal line towards the second downstream tank (pa):	$\Delta P_{11} = \frac{f_3 \cdot A_{e6}}{S_{r2}^3} \cdot \frac{\rho \cdot Q_{r2}^2}{2}$
Losses exiting the line in the second downstream tank (pa):	$\Delta P_{12} = \frac{\xi_3 \cdot \rho \cdot \left(\frac{Q_{r2}}{S_{r2}}\right)^2}{2}$

Constraints:

Y-branch water flow equality	$Q_{r1} + Q_{r2} = Q_0$ $Q_{r1} = Q_{r2}$
Downstream tubes section equality	$S_{r1} + S_{r2} = S$
Total manometric head	$H = \frac{P_{aval} - P_{amont}}{\rho \cdot g} - (H_w + H_a) + H_{r1} + \frac{\Delta P}{\rho \cdot g}$
Downstream energy balance	$\Delta P_5 + \Delta P_6 + \Delta P_7 == \Delta P_8 + \Delta P_9 + \Delta P_{10} + \Delta P_{11} + \Delta P_{12} + H_{r2} \cdot \rho \cdot g$
No cavitation phenomenon	$NPSH_a < NPSH_r$

C Bootstrap model

Constants names and values:

Flying altitude (m):	$Z = 10500$
Calorific capacity difference (J/kg/K):	$r = 287$
Mass capacity ratio (-):	$\tau = 10$
Plate conductivity (W/m/K):	$k_p = 20$
Plate thickness (m):	$t_p = 0.001$
Mass flow (kg/s):	$q = 0.7$
Isentropic efficiency of the turboreactor's diffuser (-):	$\eta_{TRd} = 0.9$
Compression ratio of the turboreactor (-):	$TC_{TR} = 8$
Isentropic efficiency of the turboreactor's compressor (-):	$\eta_{TRc} = 0.8$
Isentropic efficiency of the compressor (-):	$\eta_c = 0.75$
Isentropic efficiency of the coupling shaft (-):	$\eta_{AT} = 0.95$
Isentropic efficiency of the turbine (-):	$\eta_t = 0.8$
Heat capacity ratio (-):	$\gamma = 1.4$
Mach number (-):	$M = 0.8$

Design Variables names, domains and precisions:

Width of the exchanger (m):	$L_x \in [0.1..1]: p(L_x) = 10^{-2}$
Spacing between plates in the exchanger (m):	$r_h \in [0.001..0.1]: p(r_h) = 10^{-3}$

Auxiliary variables names and domains:

Temperature between the compressor and the exchanger (K):	$T_2 \in [0..1000]$
Temperature between the exchanger and the turbine (K):	$T_3 \in [0..1000]$
Temperature after the turbine (K):	$T_4 \in [230..500]$
Pressure between the compressor and the exchanger (pa):	$p_2 \in [0..10000000]$
Pressure between the exchanger and the turbine (pa):	$p_3 \in [0..10000000]$
Pressure after the turbine (pa):	$p_4 \in [0..10000000]$
Mass flow in the bootstrap (kg/s):	$q \in [0..1]$

Functional Variables names and definition:

Length of the exchanger (m):	$L_y = L_x$
Height of the exchanger (m):	$L_z = 0.25 \cdot L_x$
Temperature of the atmosphere (K):	$T_a = 288.2 - 0.00649 \cdot Z$
Pressure of the atmosphere (pa):	$p_a = 101290 \cdot \left(\frac{T_a}{288.08}\right)^{5.256}$
Temperature between the diffuser and the compressor of the turboreactor (K):	$T_0 = T_a \cdot \left(1 + \frac{M^2 \cdot (\gamma - 1)}{2}\right)$
Pressure between the diffuser and the compressor of the turboreactor (pa):	$p_0 = p_a \cdot \left(\eta_{TRd} \cdot \left(\frac{M^2 \cdot (\gamma - 1)}{2} + 1\right)\right)^{\frac{\gamma}{\gamma - 1}}$
Temperature between the turboreactor and the compressor (K):	$T_1 = T_0 \cdot \left(1 + \frac{1}{\eta_{TRc}} \cdot \left(\left(\frac{p_1}{p_0}\right)^{\frac{\gamma - 1}{\gamma}} - 1\right)\right)$
Pressure between the turboreactor and the compressor (pa):	$p_1 = TC_{TR} \cdot p_0$
Porosity (-):	$\sigma = \frac{r_h}{(r_h + t_p)}$

Reynolds number (-):	$Re = \frac{4 \cdot r_h \cdot G}{\mu}$
Prandtl number (-):	$Pr = 0.825 - 0.00054 \cdot T_2 + 5 \cdot 10^{-7} \cdot T_2^2$
Nusselt number (-) piecewise definition :	$Re \in [0, 2100] \rightarrow Nu = 1.86 \cdot \left(\frac{Pr \cdot Re \cdot 2 \cdot r_h}{L_x}\right)^{0.33}$ $Re \in [2100, 8000] \rightarrow Nu = 0.116 \cdot (Re^{0.66} - 125) \cdot Pr^{0.33}$ $Re \in [8000, 10000] \rightarrow Nu = \frac{10000 - Re}{10000 - 8000} \cdot 0.116 \cdot (Re^{0.66} - 125) \cdot Pr^{0.33} + \frac{Re - 8000}{10000 - 8000} \cdot 0.023 \cdot Re^{0.8} \cdot Pr^{0.33}$ $Re \in [10000, 1000000] \rightarrow Nu = 0.023 \cdot Re^{0.8} \cdot Pr^{0.33}$
Fanning factor (-) piecewise definition:	$Re \in [0, 2100] \rightarrow f = 16 \cdot Re^{-1}$ $Re \in [2100, 100000] \rightarrow f = 0.10512 \cdot Re^{-0.243}$ $Re \in [100000, 10000000] \rightarrow f = 0.04234 \cdot Re^{-0.164}$
Air viscosity (kg/m/s):	$\mu = -1.075 \cdot 10^{-5} - 2.225 \cdot 10^{-9} \cdot T_2 + 1.725 \cdot 10^{-6} \cdot \sqrt{T_2}$
Air thermal conductivity (W/m.K):	$\lambda = ((-2.620052386818974 \cdot 10^{-6}) \cdot \left(\frac{T_3 + T_2}{2}\right)^2 + (9.169307749941458 \cdot 10^{-3}) \cdot \left(\frac{T_3 + T_2}{2}\right) + 1.075874105919108 \cdot 10^{-1}) \cdot (10^{-2})$
Air density between the turboreator and the compressor (kg/m3):	$\rho_1 = \frac{p_1}{r \cdot T_1}$
Air density between the compressor and the exchanger (kg/m3):	$\rho_2 = \frac{p_2}{r \cdot T_2}$
Air density between the exchanger and the turbine (kg/m3):	$\rho_3 = \frac{p_3}{r \cdot T_3}$

Number of transfer units (-):	$Nut = \frac{H \cdot A}{q \cdot C_p}$
Exchanger efficiency (-):	$\epsilon = 1 - e^{\tau \cdot Nut^{0.22} \cdot (e^{\frac{-1}{\tau} \cdot Nut^{0.78}} - 1)}$
Exchanger inlet pressure loss	$K_e = ((-0.00496672650332) \cdot \sigma^2 +$ $(0.00113607587171) \cdot \sigma +$ $(-0.00001379297260)) \cdot \ln(Re)^2 +$ $((0.06612031387891) \cdot \sigma^2 +$ $(0.03340063900613) \cdot \sigma +$ $(-0.00178687092114)) \cdot \ln(Re) +$ $(0.96233612367662) \cdot \sigma^2 +$ $(-2.55595501972796) \cdot \sigma +$ $1.01310287017856)$
coefficient (-):	
Exchanger outlet pressure loss	$K_c = ((0.00505236835109) \cdot \sigma^2 +$ $(-0.00414707431984) \cdot \sigma +$ $(0.00347507173062)) \cdot \ln(Re)^2 +$ $((-0.08548307647633) \cdot \sigma^2 +$ $(0.06740608329495) \cdot \sigma +$ $(-0.09241949837272)) \cdot \ln(Re) +$ $(-0.18282301765817) \cdot \sigma^2 +$ $(-0.17962391485785) \cdot \sigma +$ $1.00333194877608)$
coefficient (-):	
Mass velocity (kg/m ² /s):	$G = \frac{q}{A_f}$
Exchange surface (m ²):	$A = \frac{L_x \cdot L_y \cdot (L_z - 2 \cdot r_h - t_p)}{r_h + \frac{t_p}{2}}$
Flowing section (m ²):	$A_f = L_y \cdot L_z$
Convective transfer coefficient (W/m ² /K):	$h = \frac{Nu \cdot \lambda}{r_h}$
Global heat transfer coefficient (W/m ² /K):	$H = \frac{1}{\frac{1}{h} + \frac{2 \cdot t_p}{k_p}}$
Pressure loss in the exchanger (pa):	$\Delta_{pe} = \left(\frac{G^2}{2 \cdot \rho_2}\right) \cdot (K_c + 1 - \sigma^2) + f \cdot$ $\left(\frac{A}{A_f}\right) \cdot \left(2 \cdot \frac{\rho_2}{\rho_2 + \rho_3}\right) + (K_e + \sigma^2 - 1) \cdot \left(\frac{\rho_2}{\rho_3}\right)$

Exchanger volume (m3):	$V = L_x \cdot L_y \cdot L_z$
Plate volume:	$V_p = \frac{A}{2} \cdot t_p$
Air flowing speed in the exchanger (m/s):	$C = \frac{q}{A_f \cdot \rho_2}$
Iron plate mass (kg):	$m_e = V_p \cdot 7800$

Constraints:

Compressor energy conservation:	$\eta_c \cdot \left(\frac{T_2}{T_1} - 1\right) = \left(\frac{p_2}{p_1}\right)^{\frac{\gamma}{\gamma-1}} - 1$
Coupling shaft energy conservation:	$(T_2 - T_1) = \eta_{AT} \cdot (T_3 - T_4)$
Turbine energy conservation:	$1 - \frac{T_3}{T_4} = \eta_t \cdot \left(1 - \left(\frac{p_3}{p_4}\right)^{\frac{\gamma-1}{\gamma}}\right)$
Exchanger pressure loss:	$\Delta_{pe} = p_2 - p_3$
Exchanger efficiency	$\epsilon = \frac{T_2 - T_3}{T_2 - T_0}$

1 Figures

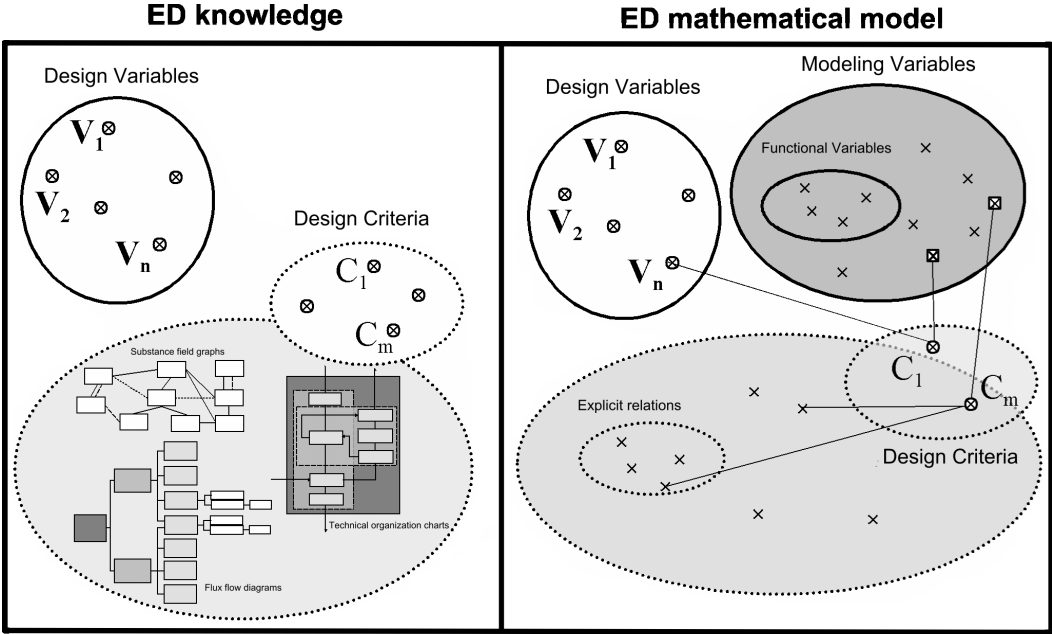


Fig. 1. Variables kind in the Embodiment Design phase.

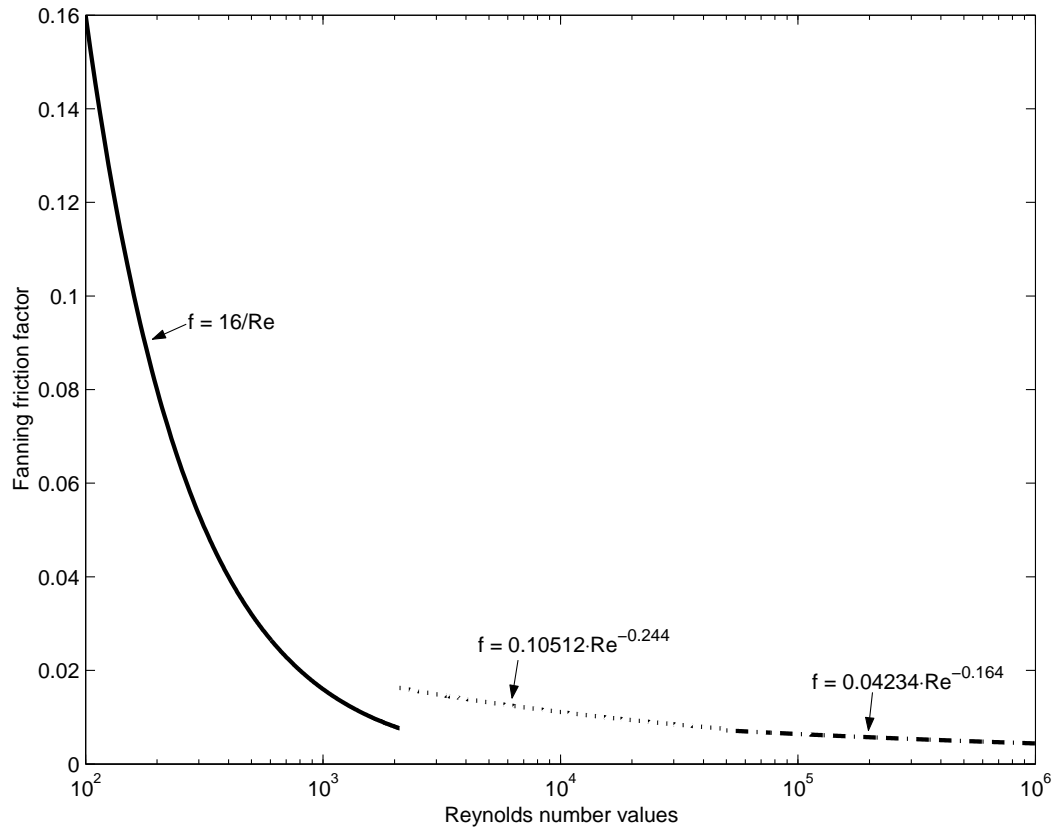


Fig. 2. Friction factor as a function of Reynolds number.

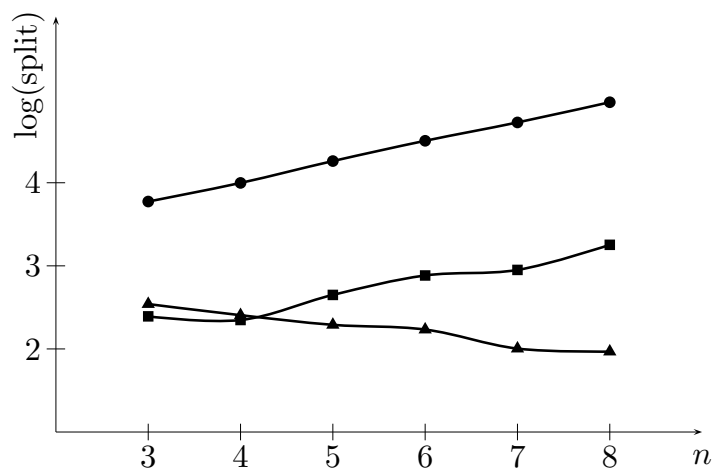


Fig. 3. Search heuristics for functional variables.

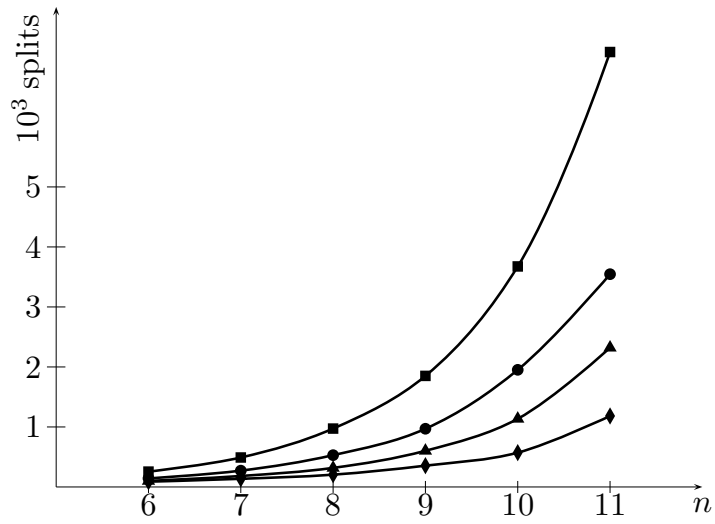


Fig. 4. Search heuristics for functional variables.

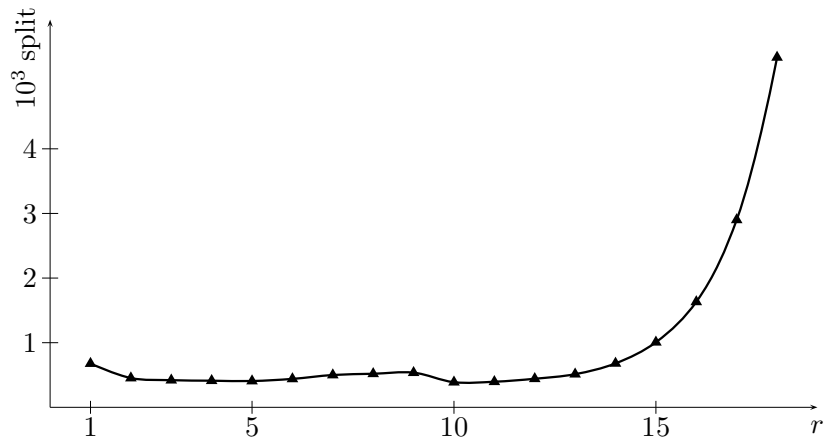


Fig. 5. Search heuristics for auxiliary variables.

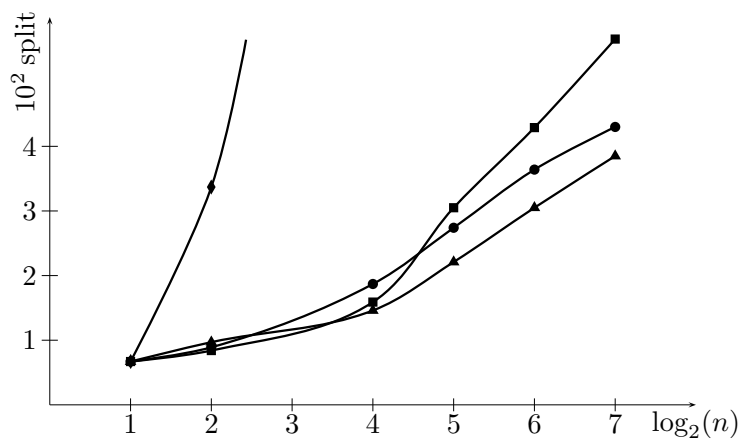


Fig. 6. Search heuristics for piecewise constraints.

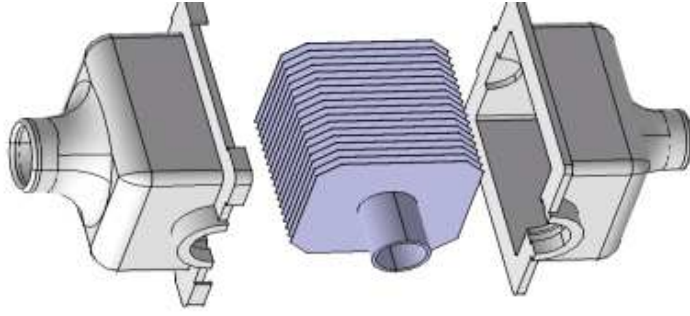


Fig. 7. Batch-exchanger.

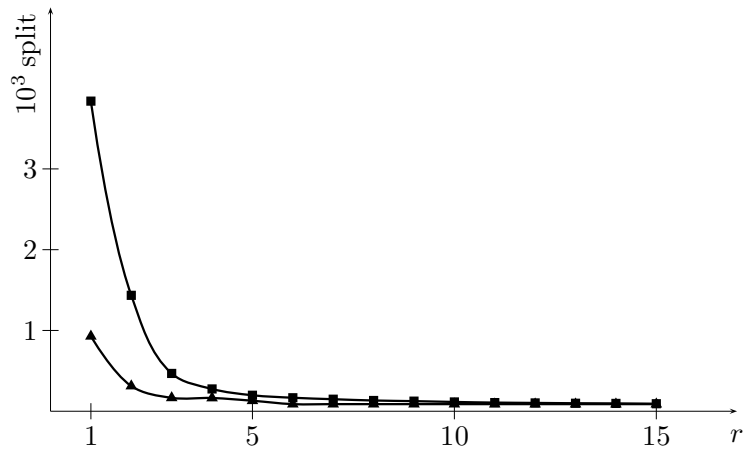


Fig. 8. Solving the batch-exchanger problem.

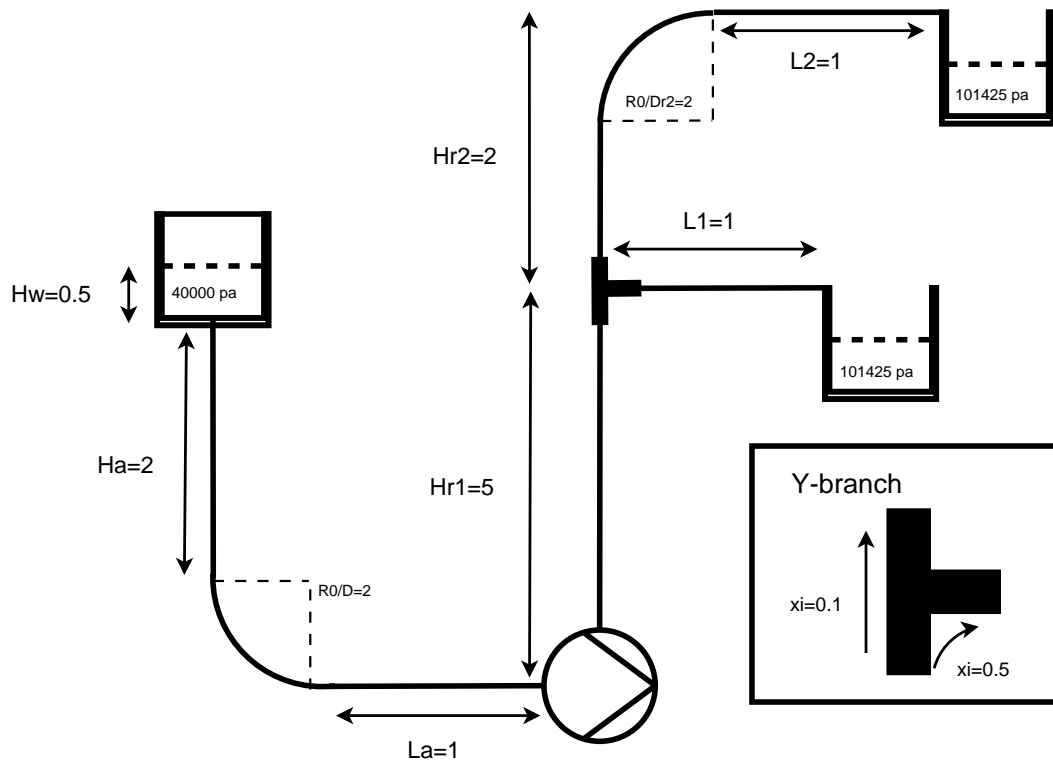


Fig. 9. Pump and tanks water circuit.

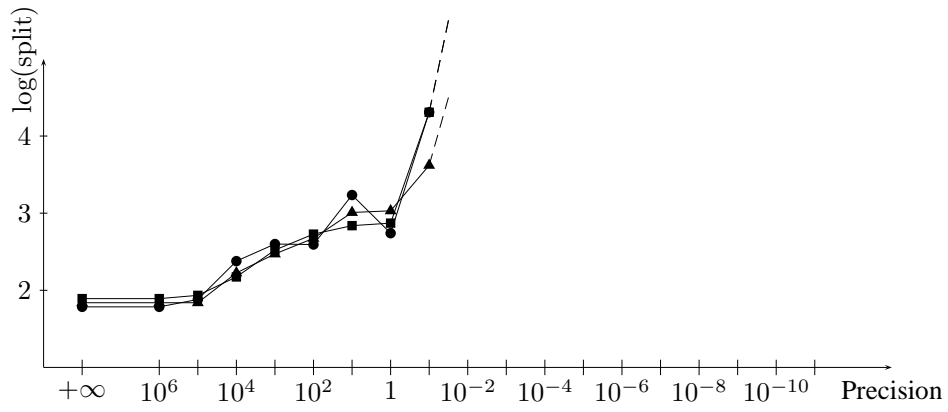


Fig. 10. Solving the pump problem with varying precision on functional variables.

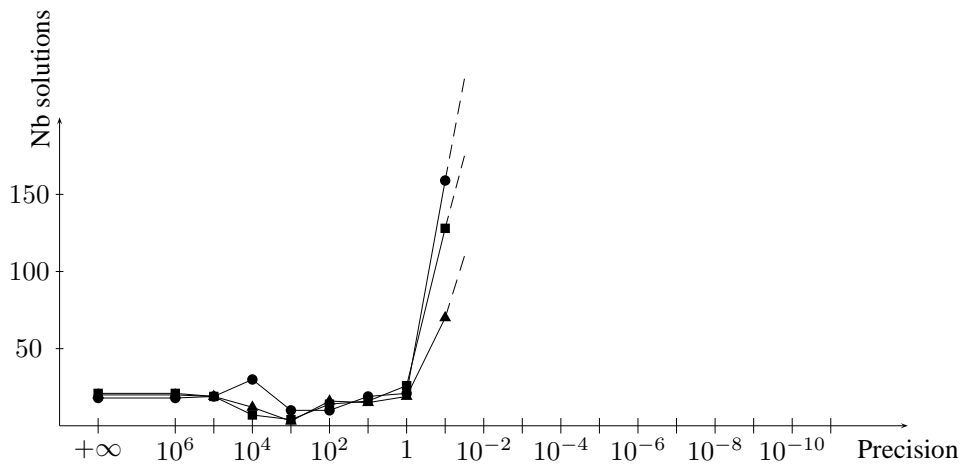


Fig. 11. Solving the pump problem with varying precision on functional variables.

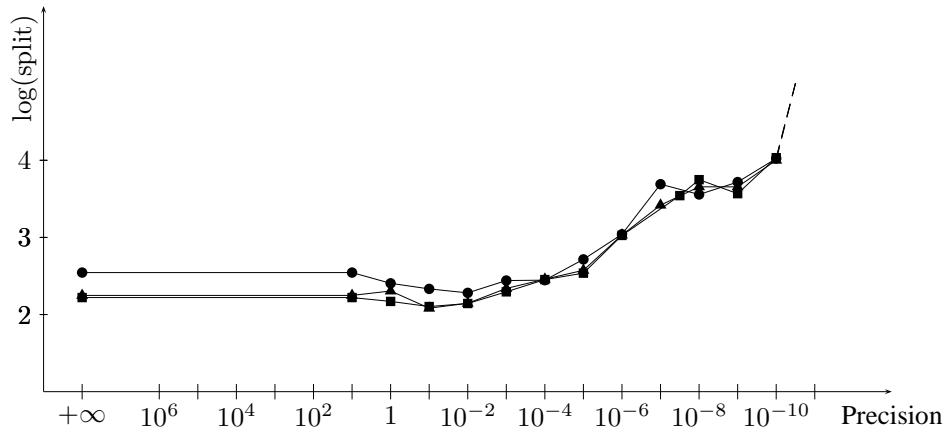


Fig. 12. Solving the pump problem with several fixed and varying precisions on functional variables.

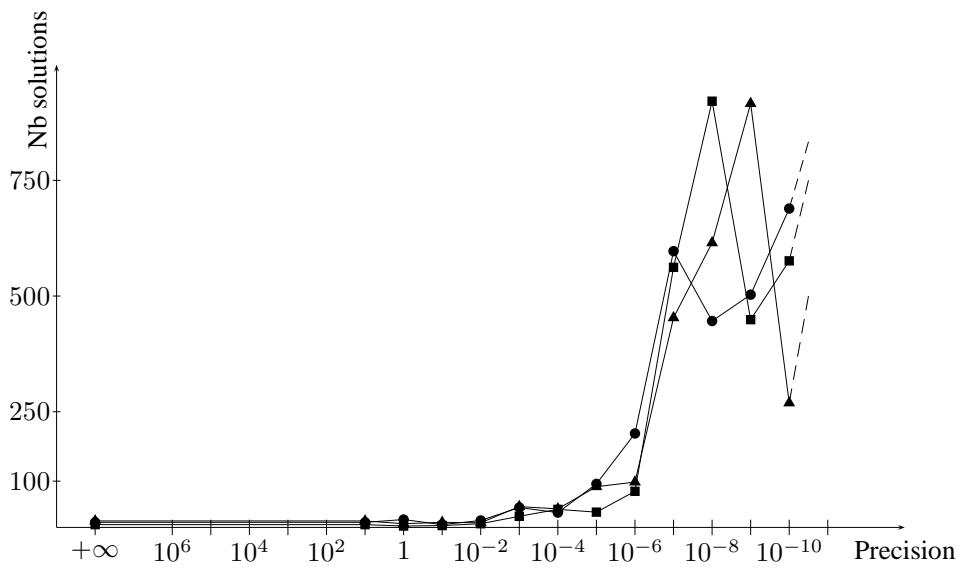


Fig. 13. Solving the pump problem with several fixed and varying precisions on functional variables.

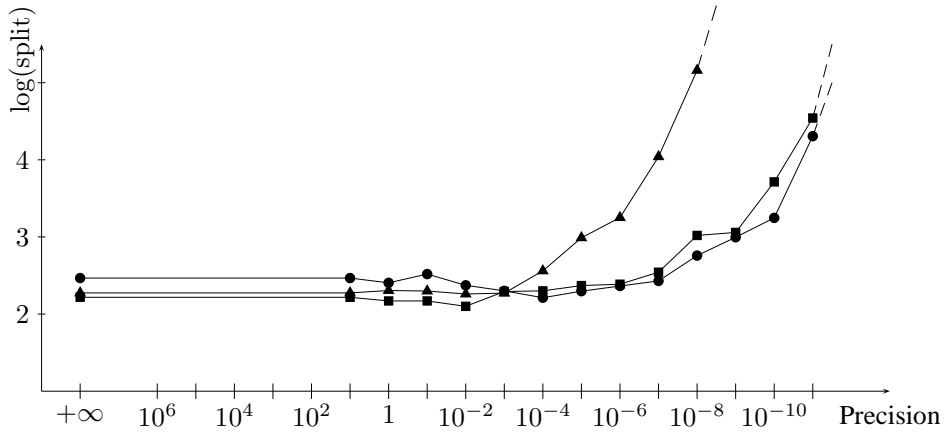


Fig. 14. Solving the pump problem with more fixed and fewer varying precisions on functional variables than those shown in figure 12 and 13.

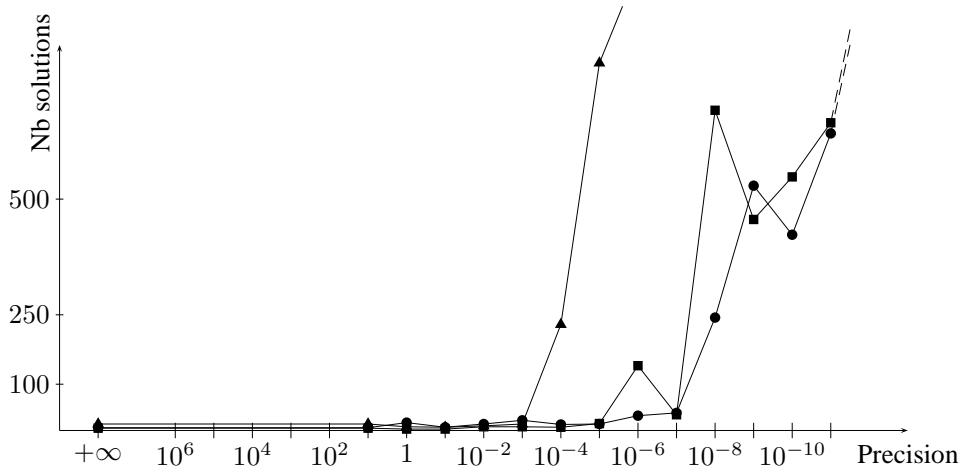


Fig. 15. Solving the pump problem with more fixed and fewer varying precisions on functional variables than those shown in figure 12 and 13.

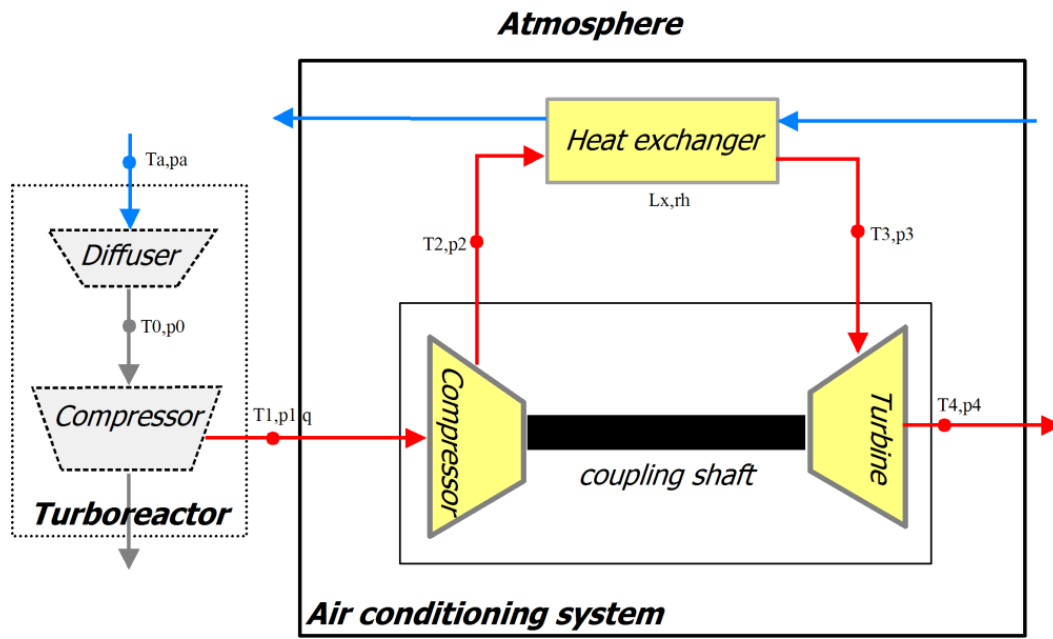


Fig. 16. Bootstrap flux flow diagram in an aircraft.