Compatibility Measurement in Collaborative Conceptual Design

D. KIM, A. BUFARDI, P. XIROUCHAKIS (2)
Institute of Production and Robotics
Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015, Switzerland.

Abstract
A product consists of various sub-functions elaborated by alternative design principles which results in many combinations. In collaborative design, designers expect that their preferences over the combinations be respected during the combination process of sub-functions, resulting in compatible combinations. We formulate the selection of compatible combinations as a combinatorial problem by: (i) defining a design principle as a list of variables with their domain and direction of designer's preferences, (ii) considering constraints among design principles, (iii) characterizing the compatibility level of combinations. A vacuum cleaner is considered with dust bag, plastic bin or cyclonic design principles for the dust collection sub-function.

Keywords:
Conceptual Design, Evaluation, Synthesis

1 INTRODUCTION

Numerous studies have been conducted in developing a conceptual design method [1] because most of the life cycle cost of a product can be pre-determined by the end of the preliminary design phase.

A product is designed to fulfil an overall function that can be often divided into a number of sub-functions, so that they compose a functional hierarchy. Each design of a sub-function can be considered as a partial design problem itself. In other words, the systematic combination of individual sub-functions results in a function structure representing the overall function [2] The next step after the establishment of a function structure is the search for possible design principles (DPs) for sub-functions that necessarily involves human creativity, namely, idea generation. The chosen DPs are then combined so as to elaborate the overall function. The large number of combinations (design alternatives) inevitably entails selecting the best ones by validation of the feasibility (i.e. design alternatives should satisfy functional constraints and other specifications) and by evaluation of optimality or satisfiability (i.e. selecting better design alternative(s) against specified technical and economical criteria). The selected design alternative is detailed during the subsequent design phase to find an optimal design solution, which maximizes the designer's expectation.

The series of refinements through decomposition and re-composition has its own advantages and pitfalls. By a suitable decomposition, once an original complex overall function is divided into a set of smaller and more manageable sub-functions, DPs for each sub-function can then be elaborated separately to some degree. On the other hand, because of the existence of many choices for DPs of a sub-function, many combinations are possible, and thus a new hard problem is raised during the best design alternative(s) selection stage.

In this paper, we explore the difficulty of managing a large number of possible combinations of DPs and focus on how to systematically sort out the more suitable combinations from the unsatisfactory ones, especially within the context of collaborative design paradigm. An overview of the procedure is provided along with a robotic vacuum cleaner (RVAC) design example.

2 COMBINATION OF DESIGN PRINCIPLES

2.1 Compatible combinations search

In conventional conceptual design, DPs are usually expressed in textual descriptions, normative symbols, or rough sketches. Therefore, the compatibilities between DPs have been validated not by a formal approach, but merely by discussion based on designers' experience and used to be characterized in a black and white manner (e.g. good/bad, +/-, etc.). Morphological matrices or selection charts have been widely used in most textbooks of engineering design (e.g. [2]). Figure 1 shows a morphological matrix assuming that the overall function consists of *n* sub-functions, each of them having *m(i)* DPs where *i*=1,2 ... *n*.

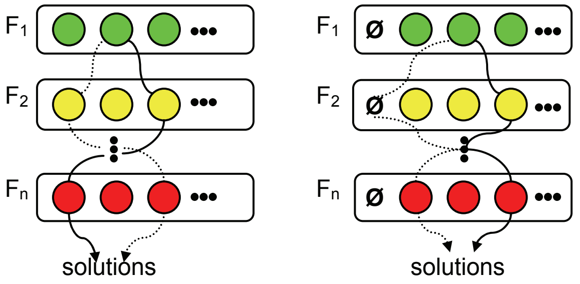
Sub-functions		Alternatives (DPs)					
		1	*2*	...	*j*	...	*m(i)*
1	F_1	A_{11}	A_{12}	...	A_{1j}	...	$A_{1m(1)}$
2	F_2	A_{21}	A_{22}	...	A_{2j}	...	$A_{2m(2)}$
⋮	⋮	⋮	⋮		⋮		⋮
i	F_i	A_{i1}	A_{i2}	...	A_{ij}	...	$A_{im(i)}$
⋮	⋮	⋮	⋮		⋮		⋮
n	F_n	A_{n1}	A_{n2}	...	A_{nj}	...	$A_{nm(n)}$

Figure 1: A morphological matrix (the shading in the cells indicates a compatible combination of DPs).

In a collaborative design environment, on the other hand, DPs are no longer represented just by a rough idea and a sketch. When we develop a new product, if we look at each sub-function itself, we can often find that several DPs for its implementation have already been invented and well-elaborated. When the properties of DPs are well-known and quantitatively elaborated, it is possible to rule out the incompatible combinations with the help of mathematical methods (e.g. constraint satisfaction techniques) although it is hard to affirm that such a method does guarantee optimal solutions and is reasonable in terms of computation time [2]. Nevertheless, little attention has been given to the measurement of compatibility of DPs. In this paper, we consider the compatibility of DPs caused by product life cycle constraints (see Figure 4-(b)). As discussed in [3], interaction of design principles does not preclude the possibility that corresponding functions are satisfied independently.

2.2 Product configuration and catalogue selection

It is worthy to compare the combination problem of DPs with the conventional configuration or catalogue selection problems considering their similar problem structure. In general, a fixed configuration of sub-functions is given in catalogue selection problems, i.e. $F_1 \times F_2 \times \ldots \times F_n$. In configuration design problems, however, any configuration of sub-functions, e.g. F_1, F_2, F_3, $F_1 \times F_n$, $F_1 \times F_2 \times F_3$ ..., can be a valid solution as shown in Figure 2-(b). For example, in a car configuration task, the sub-function, 'sunroof' can be included or excluded depending on design requirements [4]. As compared with the combination process of DPs, the configuration design problem can be assimilated to the status in which a function structure has not yet been specified sufficiently.



$$T_{\text{catalogue}} = \prod_{i}^{n} m(i) \qquad T_{\text{configuration}} = \left(\prod_{i}^{n} \left(m(i) + 1 \right) \right) - 1$$

(a) catalogue selection (b) configuration

T: number of all possible combinations of DPs
n: number of sub-functions
m(i): number of alternative DPs for sub-function *i*
Ø: the sub-function is excluded from the configuration
The empty combination '**ØxØxØx** ... **xØ**' is excluded from $T_{\text{configuration}}$

Figure 2: Comparison of product configuration and catalogue selection.

2.3 Constraints and Preferences

In conventional design catalogues, characteristics of each sub-function are described by a set of design variables/attributes. Some of them are considered for configuration such that they enable the functional interaction of the sub-functions within a function structure. We will call this set of design variables/attributes **local Design Output** (DO, for short) in order to reflect the characteristics of collaborative design. Harmer et al. [5] assumed in their component selection problem that the design of a product (or component) might involve satisfying a number of interrelated design requirements, but a suitable product can be selected from a catalogue by defining a few parameters (DOs). The values of DOs are restricted by the **design constraints** that are usually represented as relationships among DOs or restrictions/bounds on them. From this viewpoint, conflicts between DPs can be seen as violations of constraints, and can be resolved by finding alternative DPs so as to satisfy those constraints.

So far, we have discussed feasibility against design constraints. In a collaborative design environment, each sub-function can be seen as a component that is developed in parallel by different design teams where each team is primarily focusing on its own interests and objectives and thereby each team has different preferences not only on its DPs but also more fundamentally on the values of the DOs. Therefore, each design team has preferred values in the domain of a DO which meet their criteria better. For example, the suction motor design team may want to increase the suction power requiring sufficient battery capacity but it naturally requires a larger size housing design while the housing design team may prefer a smaller size housing design (see section 3.2 for more details). In other words, conflicts in collaborative design are not only due to the violation of constraints themselves, but also due to the different preferences of the design teams.

2.4 Collaborative conceptual design scenario

Considering design constraints and preferences of design teams described in the previous sections, the overall workflow of collaborative conceptual design can be illustrated as shown in Figure 3.

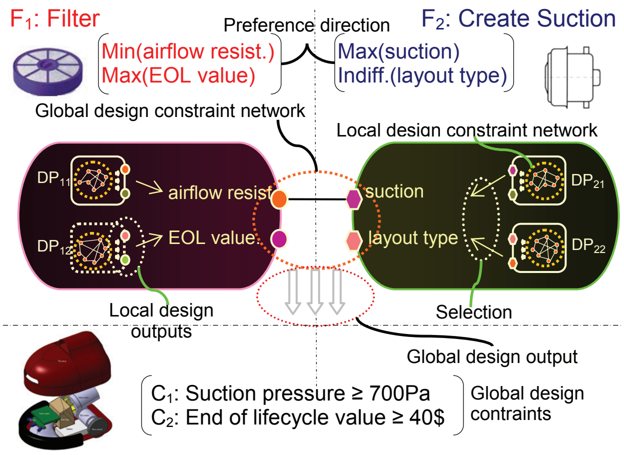


Figure 3: The workflow of constraints and preferences based collaborative conceptual design.

A product consists of a set of sub-functions designed by different design teams. They have a different set of preferences. Each DP has a local design constraint network that is a labelled graph where nodes and arcs mean local design variables and local design constraints respectively. When the values of local design variables are specified under local design constraints, each DP outputs its DOs. It is assumed that these local Constraint Satisfaction Problems (CSPs) are solved by local design teams in advance, and thus the values of DOs are given (see Figure 4). Because each design team may have a set of approximate values for a design variable based on the designer's past experience or on other rational reasons [6], each DO has a set of admissible and achievable values which can be modelled by either a **monotonic interval** or a set of **discrete choices**. Therefore, each combination of DPs creates a global design constraint network having a different set of domains for the DOs. We are interested to determine better DP combination(s) considering global design constraints and design team's preferences on the DOs.

3 PREFERENCE BASED COMBINATION OF DESIGN PRINCIPLES

3.1 Design catalogue for a robotic vacuum cleaner

Figure 4 shows the simplified design catalogue for a RVAC considering four main sub-functions. The value in the preference direction column of DO_{41}, 'Min' means that a lower value, 'flat' is preferred to a higher value, 'top'. 'Indiff.' in DO_{22} means the design team has no preference among the options. All design constraints C_i considered are binary for simplicity. It is assumed that the values for the energy consumption (Watt) of F_2 are approximated (i.e. 1 Watt $\cong$ 0.1×suction pressure in Pa) in C_4 because energy consumption is proportional to suction pressure.

F_i	DP_{ij}		Pref.
F_1: Filter	Regular	HEPA	
DO_{11}: Airflow resistance(%)*	[3...7]	[5...12]	Min
DO_{12}: EOL value($)	[5...15]	[15...25]	Max
F_2: Create suction	Type A	Type B	
DO_{21}: Suction (Pa)	[600...900]	[850...1200]	Max
DO_{22}: layout type (axis-air dir.)	[H-H, V-V]	[V-H]	Indiff.
F_3: Provide Energy	Ni-Cd	Li-ion	
DO_{31}: Capacity(mAh×V)	[15...30]	[20...50]	Max.
F_4: Collect dust & Housing	Dust bag or plastic bin	Cyclone	
DO_{41}: Housing type	[flat, semi]	[top]	Min.
DO_{42}: EOL value($)	[5,10,15,20]	[15,20,25]	Max

(Pref.: preference direction, * : % of suction pressure)

(a) Domain and direction of designer's preferences

C_1: DO_{21} × (100%-DO_{11}) ≥ 700Pa (suction)
C_2: DO_{12} + DO_{42} ≥ 40$ (end of lifecycle value)
C_3: if DO_{22} = V-V or V-H then DO_{41} = top (topological)
C_4: DO_{31} ≥ 0.5×0.1×DO_{21} (Operating time ≥ 0.5hr.)

(b) Global design constraints

Figure 4: Design catalogue for a RVAC.

3.2 Design issues for RVACs

The suction motor capacity and the filtration method play crucial roles in the efficiency of a vacuum cleaner. The amount of fine dust particles (usually less than 0.5μm) penetrating through a vacuum cleaner and the exhaust odours determine the quality of the air we will breathe after cleaning. Therefore, many commercial vacuum cleaners have been equipped with a final high-efficiency particulate air (HEPA) filter. However, this high-density filtration leads to an interference with the airflow so as to have a negative effect on the suction motor. In other words, vacuum cleaners are supposed to filter off impurities as much as possible. As a result, the suction motor should have more power to compensate for the loss of airflow but it generates more noise and requires a larger housing design and sufficient battery capacity.

3.3 Quantitative compatibility measuring

In [7], in order to measure the compatibility between two DPs, we defined the compatibility value by incorporating the REVISE algorithm [8] or interval arithmetic with the preferences of design teams to the DOs. The background notion of the proposed method relies on the fact that as one DP restricts less (by a set of constraints) a preferred region within the domain of a DO of another principle, the two DPs can be seen to be more or less compatible with each other. By summing up these compatibility values of all pairs of DPs in a combination, we calculated indirectly the overall compatibility score for the combination, and finally found the most compatible combination(s). The pairwise evaluation of compatibility in [7] can be useful when: (i) the domains of DOs have not sufficient common values such that search efficiency does not change by simply merging the domains, (ii) design constraints are not so interrelated and complex, (iii) there exist design constraints which clearly specify the compatibility of pairs of DPs such as C_3. While the pairwise evaluation of compatibility is relatively intuitive and quick, it tends to overlook interrelationships among all the sub-functions. In place of the simple REVISE algorithm, this paper presents a direct search method based on the Arc Consistency (AC) algorithm [8] that is the iterative application of the REVISE algorithm until there is no change on the domain of DOs.

Step 1: Union of domains

In general, a CSP consists of a finite set of variables, their domains and constraints. Every variable has a domain that is a set of possible values. In our combination problem, every DO has a set of domains. Therefore, for the i^{th} sub-function, the domain of the k^{th} DO of the j^{th} DP are merged all together into a new unified domain UDO_{ik}, which can be explained by:

$$UDO_{ik} \equiv \bigcup_{j=1}^{m(i)} domain(DO_{ik}(DP_{ij}))$$

$$where\ i=1,2,\ ...,\ n,\ j=1,2,\ ...,\ m(i),\ k=1,2,\ ...,\ q(i)$$

$$e.g.\ UDO_{11} \equiv domain(DO_{11}(DP_{11})) \cup domain(DO_{11}(DP_{12})) \equiv [3...7] \cup [5...,12] \equiv [3..,12]$$

Note that a set of vectors with elements, UDO_{ik} for all *i* and *k* is denoted as **UDO**. The new domains, **UDO** corresponding to our RVAC design are shown in Table 1

F_1	F_2	F_3	F_4
UDO_{11} [3...12]	UDO_{21} [600...1200]	UDO_{31} [15...50]	UDO_{41} [flat, semi, top]
UDO_{12} [5...25]	UDO_{22} [H-H, V-V, V-H]		UDO_{42} [5,10,15,20,25]

Table 1: Unified domains UDO of the RVAC design

Step 2: Representation of preference strength

Two types of DO domains need to be considered:

- **Case 1**: a set of discrete choices,
- **Case 2**: a monotonic interval.

Assign a bigger weight to a more preferred value in a UDO. Note that weights here have been normalized to add up to 1. Needless to say, more precise non-ordinal values could be specified for the weights with a more detailed analysis of the preferences.

Case 1: e.g. the values: flat, semi and top of UDO_{41}, have the normalized weights: 3/6, 2/6, and 1/6 respectively because 'flat' is preferred to 'semi' which is preferred to 'top'.

Case 2: When the interval of a DO x is given as $[I^{min}...I^{max}]$, the weight function is defined as follows:

$$w(x) = \frac{2(x - I^{\min})}{(I^{\max} - I^{\min})^2} \quad \text{if the preference direction is 'Max'}$$

$$w(x) = -\frac{2(x - I^{\max})}{(I^{\max} - I^{\min})^2} \quad \text{if the preference direction is 'Min'}$$

$$w(x) = \frac{1}{I^{\max} - I^{\min}} \quad \text{if the preference direction is 'Indiff'}$$

Clearly $\int_{I^{\min}}^{I^{\max}} w(x)dx = 1$

Step 3: Build a New single CSP (N-CSP)

Since there is no change in the given set of design constraints **C**, we can build a new single CSP having **UDO** for the new domains. It can be formally represented by a constraint network, < **DO**, **UDO**, **C** >, where **DO** is a set of DO_{ik} for all *i* and *k* (see Figure 4-(b) and Table 1).

Step 4: Consistency check

At this point, apply the consistency technique to eliminate inconsistent values that cannot take part in any potential solution of N-CSP. It is worth noting that many variant approaches for consistency techniques have been proposed in the literature on CSP. In the case of relatively simple problems, we can pursue the strong arc consistency of the constraint network as opposed to partial arc consistency techniques [9] applicable for large space problems. However, we limit the discussion to the basic scheme, and so we are not concerned with the detail that one method is more effective or not. Therefore, in this paper, we use AC-3 [8] for a basic mechanism which allows all variables and associated constraints to be revised in any order. The revised domains of **UDO** after consistency checks are shown in Table 2 and denoted as **RDO** in order to avoid notational conflict.

F_1	F_2	F_3	F_4
RDO_{11} [3...12]	RDO_{21} [796...1000]	RDO_{31} [40...50]	RDO_{41} [flat, semi, top]
RDO_{12} [10...25]	RDO_{22} [H-H, V-V, V-H]		RDO_{42} [15,20,25]

Table 2: Revised domains after consistency checks for the DOs of the RVAC design.

We now have a simpler but equivalent problem in terms of the feasible solution space than the original N-CSP. Stated another way, by pruning inconsistent values from **UDO**, only the values having more possibility to take part in a solution of N-CSP remain.

Step 5: Search the most compatible combination(s)

In order to determine the most compatible DP_{ij^*} of a sub-function F_i, we now consider the following definition:

Definition 1. If a DP of a sub-function is compatible with the other sub-functions, then the values in the domains of its DOs have more common values in the preferred region with the corresponding domains of **RDO** than the other DPs.

First, find common values (intersection) between the domain of DO_{ik} of DP_{ij} and RDO_{ik} as follows:

$$I_{ijk} \equiv domain(DO_{ik}(DP_{ij})) \cap RDO_{ik}$$

If there is no common value, we can say that the DP is not compatible with the other sub-functions. By definition 1, the compatibility value CV_{ij} of DP_{ij}, can be calculated from the sum of weights on the common values:

$$CV_{ij} = \sum_{k=1}^{q(i)} W(i,k) \text{ where } q(i)\text{:number of DOs of sub-function } i$$

$$\text{Case1: } W(i,k) = \sum_{p=1}^{r(i,k)} (w_{ikp} \times \alpha) \text{ where } \begin{cases} \alpha = 1 \text{ if } v_{ikp} \in I_{ijk} \\ \alpha = 0 \text{ otherwise} \end{cases}$$

$$\begin{cases} w_{ikp}\text{: weight on } v_{ikp} \\ v_{ikp}\text{: value } p \text{ of } UDO_{ik} \quad p=1,2,\ldots,r(i,k), \\ r(i,k)\text{: size of } UDO_{ik}, \text{i.e. } \|domains(UDO_{ik})\| \end{cases}$$

$$\text{Case2: } W(i,k) = \int_{I_{i,k}^{\min}}^{I_{i,k}^{\max}} (w_{i,k}(x) \times \alpha)\,dx \text{ where } \begin{cases} \alpha = 1 \text{ if } x \in I_{ijk} \\ \alpha = 0 \text{ otherwise} \end{cases}$$

$$\begin{cases} I_{i,k}^{\max}\text{: max value in the interval of } UDO_{ik} \\ I_{i,k}^{\min}\text{: min value in the interval of } UDO_{ik} \\ w_{ik}(x)\text{: weight funtion of } UDO_{ik} \end{cases}$$

We can now recognize the most compatible design principle(s) DP_{ij^*} in the sub-function *i* by searching for the DP that has the maximum compatibility value CV_{ij^*}. Finally, we can find the most compatible combination, $C^* \leftarrow DP_{1j^*} \times DP_{2j^*} \times \ldots \times DP_{nj^*}$ .The above procedure can be illustrated as shown in Figure 5.

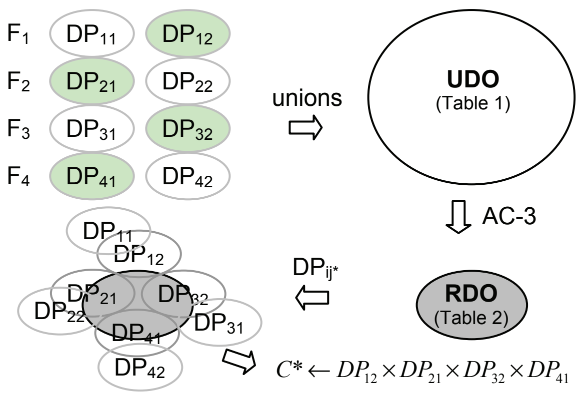


Figure 5: Illustration of the procedure to find *C**.

The compatibility values of all DPs and their calculation processes are summarized in Table 3, where the combination of HEPA, Type A, Li-ion, and dust bag or plastic bin is the best compatible combination.

Note that the pair of DP_{22} and DP_{41} is incompatible by the constraint C_3 and CV_{31} is 0. Therefore, any combination including the pair or DP_{31} is incompatible.

F_i	DP_{ij}	DO_{ik}	I_{ijk}	CV_{ij}
F_1	DP_{11}	DO_{11} DO_{12}	[3...7] [10...15]	$\int_3^7 -\frac{2(x-12)}{(12-3)^2}dx + \int_{10}^{15} \frac{2(x-5)}{(25-5)^2}dx = \frac{56}{81} + \frac{75}{400} \cong 0.88$
	DP_{12}	DO_{11} DO_{12}	[5...12] [15...25]	$\int_5^{12} -\frac{2(x-12)}{(12-3)^2}dx + \int_{15}^{25} \frac{2(x-5)}{(25-5)^2}dx = \frac{49}{81} + \frac{300}{400} \cong 1.35$
F_2	DP_{21}	DO_{21} DO_{22}	[796...900] [H-H, V-V]	$\int_{796}^{900} \frac{2(x-600)}{(1200-600)^2}dx + \frac{1+1}{3} = \frac{51548}{360000} + \frac{2}{3} \cong 0.81$
	DP_{22}	DO_{21} DO_{22}	[850...1000] [V-H]	$\int_{850}^{1000} \frac{2(x-600)}{(1200-600)^2}dx + \frac{1}{3} = \frac{975}{3600} + \frac{1}{3} \cong 0.60$
F_3	DP_{31}	DO_{31}	**ø**	0
	DP_{32}	DO_{31}	[40...50]	$\int_{40}^{50} \frac{2(x-15)}{(50-15)^2}dx = \frac{750}{1225} \cong 0.61$
F_4	DP_{41}	DO_{41} DO_{42}	[flat, semi] [15, 20]	$\frac{3+2}{1+2+3} + \frac{2+3}{1+2+3+4+5} \cong 1.17$
	DP_{42}	DO_{41} DO_{42}	[top] [15,20,25]	$\frac{1}{1+2+3} + \frac{3+4+5}{1+2+3+4+5} = 0.97$

Table 3: Compatibility values of all DPs.

4 CONCLUSION

This paper presented a systematic method to quantitatively evaluate the compatibility of a design principle with the other sub-functions, which is achieved by incorporating designers' preferences about the design outputs into the consistency technique. Given a design catalogue, the method proposes, without an exhaustive search, suitable combinations of design principles of sub-functions on the basis of both their feasibility with respect to the design constraints and the designers' preferences.

5 REFERENCES

[1] Jin, Y., Li, W., Lu, S., 2005, A Hierarchical Co-Evolutionary Approach to Conceptual Design, Annals of the CIRP, 54/1:155-158.

[2] Pahl, G., Beitz, W., 1988, Engineering Design - A Systematic Approach, Springer-Verlag, London.

[3] Chen, L., Lin, L., 2002, Optimization of Product Configuration Design Using Functional Requirements and Constraints, Res Eng Des, 13:167-182.

[4] Mittal, S., Falkenhainer, B., 1990, Dynamic Constraint Satisfaction Problems, Proc. of the 8th AAAI, 25-32, Boston.

[5] Harmer, Q., Weaver, P., Wallace, K., 1998, Design-LED Component Selection, CAD, 30/5:391-405.

[6] Gheorghe, R., Bufardi, A., Xirouchakis, P., 2005, Fuzzy Multicriteria Decision Aid Method for Conceptual Design, Annals of the CIRP, 54/1:151-154.

[7] Kim, D., Burfadi, A., Xirouchakis, P., 2004, Preference-Based Combinatorial Design Problem Solving in a Collaborative Environment, Proc. of the TMCE 2004, 2:903-911, Lausanne.

[8] Mackworth, A., 1977, Consistency in Networks of Relations, Artificial Intelligence, 8/1:99-118.

[9] Nadel, B., 1989, Constraint Satisfaction Algorithms, Computational Intelligence, 5:188-224.