Solving The Paradox Of Constraints In Creativity: Uncovering The Conditions Of Constraint Generativity With C-K Design Theory

Pascal Le Masson, Armand Hatchuel, Benoit Weil

Solving The Paradox Of Constraints In Creativity: Uncovering The Conditions Of Constraint Generativity With C-K Design Theory

Pascal Le Masson, Armand Hatchuel, Benoit Weil

Mines Paris PSL,

Center of Management Science -i3 UMR CNRS 9217

In: The paradox of constraints in creativity, edited by Catrinel Tromp, Robert J. Sternberg, and Don Ambrose, part of the Advances in Creativity and Giftedness series of Brill Publishers.

Abstract (150–250 words)

The literature on constraints in creativity remains inconclusive (Sternberg and Kaufman 2012) in that constraints can limit (Amabile 1998; Rosso 2014) or support (Onarheim 2012; Keupp and Gassmann 2013; Haught-Tromp 2017) creativity.

However, a new approach suggests that examination of the design process can explain how a constraint encourages or impedes creativity (Hatchuel and Klasing Chen 2017). This approach is based on advances in the theory of generativity according to C-K design theory (Hatchuel et al. 2018). C-K design theory provides a formal model of how creativity can be limited or enhanced (Hatchuel et al. 2011b), that has been experimentally confirmed (Agogué et al. 2014a; Ezzat et al. 2018; Arrighi et al. 2015; Freitas Salgueiredo and Hatchuel 2016).

In this chapter, we deepen the approach by establishing three critical conditions and mechanisms that explain how a given constraint P^* improves the generativity of a design process: we show that a constraint is all the more generative when the constraint brings knowledge that is independent of the designer's initial knowledge, when the constraints closes a 'fixation' area, and when the design process oriented by the constraint leads to uncover new knowledge independent of designer's initial knowledge.

These findings explain the variety of generative effects associated with constraints, the conditions for a constraint to be generative in a given design task, and why some constraints can unexpectedly increase or decrease designers generativity. Finally, the findings confirm the resolution of the paradox of constraints in creativity.

I. Introduction: Addressing The Issue Of Constraint Generativity With C-K Design Theory

Contemporary grand challenges (e.g., those relating to mobility, education, biodiversity, and climate) can be phrased as a need for innovation under strong constraints (e.g., constraints on financial resources, energy, materials, and competencies). One might be optimistic looking back at artistic traditions or the exploits of famous engineers (e.g., Brunelleschi's dome was constructed without scaffolding). However, optimism decreases when one considers that constraints are also said to limit innovation and creativity. Do constraints encourage or impede creativity?

The literature remains surprisingly inconclusive (Sternberg and Kaufman 2012; Damadzic et al. 2022), despite important recent contributions. Constraints were initially perceived as restricting creativity, the latter being seen as benefiting from free imagination and freely available resources (Johnson-Laird 1988; Amabile 1998; Amabile et al. 2005). More recently, authors have increasingly acknowledged that constraints might also support creativity, as observed in experimental works (Agogué et al. 2014a; Ezzat et al. 2018; Ezzat et al. 2017a; Ezzat et al. 2017b),, real-life studies conducted in various environments (Onarheim 2012; Rosso 2014; Keupp and Gassmann 2013; Arrighi et al. 2015; Eckert et al. 2012; Groop et al. 2015; Hatchuel and Klasing Chen 2017), and computing science (Krish 2011; Vajna et al. 2005; Arrighi et al. 2016). Exhaustive systematic reviews of the literature (Onarheim and Biskjaer 2013; Acar et al. 2018) have provided consensus on some points.

- Frequent inconclusiveness: It is well documented that constraints can limit (Amabile 1998; Rosso 2014) and support (Onarheim 2012; Keupp and Gassmann 2013; Haught-Tromp 2017) creativity . It remains difficult to clarify what makes that a constraint positively contributes to having more varied and more original ideas.

- Plausibility of an inverted U-shaped law: One might make the hypothesis of an inverted U-shaped relation between the level of a constraint and its generative effect (Acar et al. 2018; Rosso 2014). A “light” constraint would slightly contribute to creativity, a stronger one would strongly contribute and very strong constraint would impede creativity (Gillier et al. 2018).
- Constraint aporia: New cognitive approaches (Haught-Tromp 2017; Sternberg and Kaufman 2012) are being adopted in studying the deep relations between constraints, knowledge, and creativity to address the apparent aporia (Johnson-Laird, 1988) of how a constraint, which would at the same time *restricts* the solution space (because it constraints the exploration in a specific direction) but would also enlarge it since a generative constraint is supposed to lead to solutions that were not accessible without it. Hence the aporia: how can a constraint simultaneously restrain and open a solution space?

The present paper focuses on one particular point in shedding light on the effect of constraints on creativity and innovation: how constraints contribute to *generativity* in a *given* design *task*. This formulation corresponds to three assumptions that need clear statements:

Assumption 1: We focus on *generativity*, and even radically original generativity – seen as the generation of novel, original propositions. Focusing on generativity, we want to avoid possible misinterpretation related to ‘innovation’ or ‘creativity’: a) Generativity and innovation: innovation relates both to novelty and market success – hence innovation might include rigorous product development processes; as a consequence a constraint could thus contribute to innovation by enabling efficient development of one single product, which might be far from generativity - in this case one will have a relationship between constraint and innovation but not necessarily between constraint and generativity, ie only the ‘novelty’ facet of innovation. b) Generativity vs creativity: creativity is often defined using Amabile’s definition of being novel and useful. It is thus unclear to which facets constraints contribute: constraints can be considered as contributing more to usefulness than to originality and one could have a relationship between constraint and creativity (usefulness) but not necessarily between constraint and generativity. In this paper we focus on this relationship between constraint and generativity, the ie only the 'originality' facet of creativity. c) For more rigor, we will rely more precisely on a *narrow* definition of generativity: recent reviews on generativity (Thomas and Tee 2021) have shown that generativity can be either rule-based (i.e., model-based, deterministic, combinatorial), such as for the meaning of generativity given by Zittrain (Epstein 1999; Zittrain 2006), or generativity can be more "radical" (i.e. changing the design rules, relying on newly discovered rules…), such as for the meaning of radical originality given by Boden (e.g., (Boden 1999; Hatchuel et al. 2011a). In this chapter, we consider generativity in terms of this demanding meaning of creating *radically original* propositions. Hence our first assumption: we study how constraints relate to generativity where generativity will be associated to this generation of radically original propositions.

Assumption 2: As explained by Haught-Tromp (2017), one can distinguish between constraints relating to the design *situation* (the context where the design takes place: within a certain company, with certain competencies, within a certain budget, following organizational procedures, etc.) and constraints 'directly' linked to the design *task* (e.g., an object X with the property P has to be designed also with the property P^*). In this chapter, we only consider constraints relating to the design task. This restriction is justified by two reasons. 1) It allows us to focus on the design process itself and to solidly model the design process under constraints. 2) It allows us to focus on the critical process in which the constraint acts on design generativity. We also expect that a better understanding of the effect of the *task* constraint on generativity can contribute to the formulation of specific hypotheses of the effect of a *situation* constraint on generativity.

Assumption 3: In design, a constraint can have several meanings and roles. Here, we focus on modeling situations where *a specific design task is given* i.e., the task "design X with property P" is identified before an (additional) constraint comes into play, and the constraint that we study is *imposed on top of the design task* (i.e., the constraint P^* is added to the design task, so that the task becomes "design X with properties P *and* P^*"). In this situation we can analyze the effect of a constraint P* on a design task "design X with property P". Why this specific approach? In particular it is clear that one could consider P as a "constraint" imposed on the object X. However this would lead to a slightly different research question: the effect of imposing a constraint P on an object X – and X as such is not a design task and we could then wonder whether imposing P 'creates' a design task – this could be an interesting and relevant question[1]. Still it is not question we address in this paper where we more specifically want to discuss the effect of a constraint P* on a design task, hence comparing the design task development without the constraint P* vs with the constraint P*.

These three assumptions clarify our research topic. Given a certain design task (design X with property P), we analyze whether the same task with an additional constraint P^* (design X with properties P and P^*) increases or decreases generativity, i.e. the generation of radically

[1] The answer to this question is actually quite straightforward in design theory such as C-K design theory – there are three possibilities: i/ either P is a property that the object X can already have (X =a car, P=with wheels), in which case there is no generativity induced by this constraint P (cars with wheels exist already, no radically original proposition); ii/ or the property P is proven impossible for the object X (X=a physical object, P=with perpetual motion) in which case the constraint leads to nothing (a object with perpetual motion is proven impossible – even patent law recommends to refuse patents which claim perpetual motion), iii/ or it is undecidable whether there is an object X with property P and there begins a design process that, if successful, creates a radically original proposition.

original propositions so solve the design task X with property P. The present paper makes the following contributions to the literature. 1) We model the effect of a constraint in the design process. 2) We contribute to solving the aporia described above by showing that a constraint can (seem to) restrain the exploration space but in parallel open the door to new knowledge that acts as a resource for future exploration. 3) We establish three critical mechanisms (i.e., knowledge injection, exploration partition, and knowledge discovery) that explain how a given constraint P^* affects the generativity of a design process: in knowledge injection the constraint can bring knowledge that is independent (or not) of the designer's initial knowledge, in exploration partition the constraint can close a 'fixation' area (or not), and in knowledge discovery the design process oriented by the constraint can lead to uncover new knowledge independent of designer's initial knowledge (or not)). From these mechanisms we deduce a sufficient condition and a necessary condition for a generative constraint.

The remainder of the paper is organized as follows. We first establish the above results adopting a formal approach, based on C-K design theory. We then present several empirical results for different contexts (i.e., laboratory experiments, computer science, and real-life experiments) that confirm the theoretical prediction. We finally show how our results might be extended to other types of constraint.

II. Modeling The Generative Effect Of Constraints With Design Theory

II.1. C-K Theory And Generativity

To model the generative effect of constraints on a given design task we rely on design theory, which provides us today with advanced and well-controlled models of generativity, one of the most advanced of these models being C-K design theory (Hatchuel et al. 2011a; Hatchuel et al. 2018).

C-K theory models a design process as follows.

1) Given a knowledge base K_{ini} comprising true and/or false propositions, a concept C_0 is a proposition of the form "C_0 = there is an X with property P such that i/ XP is interpretable (i.e., X relates to knowledge $K_{ini}(X)$ in K_{ini} and P relates to $K_{ini}(P)$ in K_{ini}) and ii/ the proposition is neither true nor false in K_{ini}, i.e. it cannot be proven with K_{ini} that XP exists or that XP cannot exist. There is a disjunction between XP and K_{ini}, and XP is unknown in K_{ini}.

2) The design process is a dual expansion.

a) Expansion of K_{ini}: New propositions δK are added to the K space – this is a learning process. By construction, these propositions are true or false.

b) Partition of C_0 = XP: It is possible to add a property P_i (from the K space) to the initial concept C_0 to form the proposition XPP_i. One has to check whether XPP_i is now true (or false) or whether XPP_i is still in C. In the latter case, XPP_i is a partition of XP. XPP_i can be used to generate new knowledge, which is another expansion of K_{ini}. Note that in C-K theory, one distinguishes between two types of partition. Either P_i is deduced from $K_{ini}(X)$ (ie P_i is dependent of K_{ini}) and the partition is said to be restrictive or P_i is independent of $K_{ini}(X)$ and the partition is said to be expansive.

3) The process stops when $XPP_1 \ldots P_n$ becomes true in K. Here, $XPP_1 \ldots P_n$ is a conjunction. See Figure 1-a below for the synthesis. Note that a conjunction requires knowledge expansion δK. Brief demonstration: Suppose that a conjunction XPP1…Pn is made without expansion. The conjunction comes from $K_{ini}(X)$ and XP is actually true in $K_{ini}(X)$. However, XP is a concept, unknown in $K_{ini}(X)$. That is to say, generativity is

associated with new knowledge δK, beyond $K_{ini}(X)$[2]. A conjunction can be based on new knowledge *depending* on K_{ini}, in which case this corresponds to rule-based generativity (see above: generativity based on known design rules, that are combined and optimized); or conjunction can come from new knowledge independent of K_{ini} (newly discovered design rules – see above: generativity that corresponds to radically original propositions). In C-K terms, rule-based generativity is called restrictive generativity and the generation of radicaly new propositions is called expansive generativity (see figure 1 below).

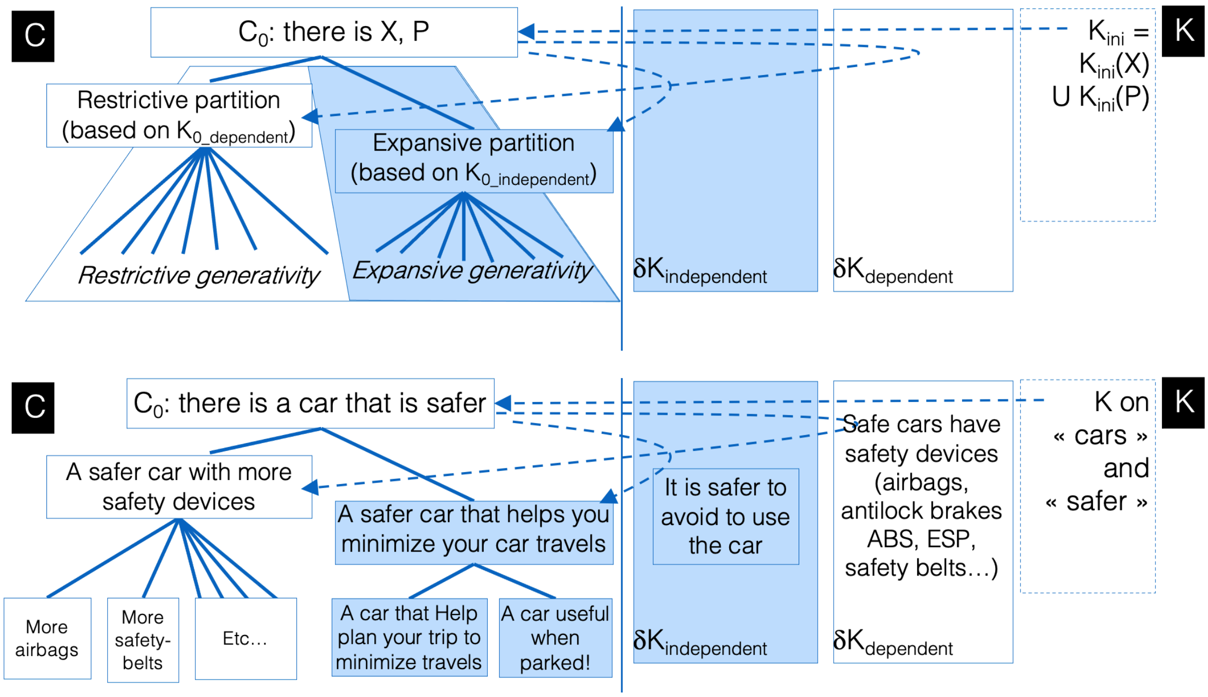

Figure 1: a representation of C-K design process (without specific constraint) (figure 1-a) and one simple example (figure 1-b). Figure 1-a: in C-space, the concepts (ideas, chimeras, etc) either **expansive** (grey-shaded area in C-space), meaning that it is based

---

[2] By construction, a conjunction is a proof of independence. XP is unknown initially, XP is not true in $K_{ini}(X)$, and thus, all X in K_{ini} are non-P. Following the conjunction $XPP_1…P_n$, there is an X with property P. Hence, there is an X with non-P and an X with P and thus X and property P are independent. *Therefore, δK contains knowledge that is indispensable for this proof of independence.*

on new knowledge, independent from K_{ini} (grey-shaded area in K-space), or **restrictive** (white area in C-space) meaning that it is based on knowledge that is dependent of K_{ini}, i.e. resulting from combination, deduction, inference,… from K_{ini} (white area in K-space). Figure 1-b: a simplified example on the concept “a safer car”.

We can now express our research topic in C-K terms: we are interested in expansive generativity caused by constraint. We will now explain how we model a constraint in C-K theory.

II.2. Analyzing The Generativity Of A Constraint In C-K Theory

Intuitively, a constraint will be added to the design process described above in C-K and this ‘constraint’ can make the process generate ‘expansive’ propositions (expansive generarivity) or ‘restrictive’ one. We know from C-K that expansive generativity is reached to the condition that the designer relies on new knowledge, independent from the initial one. Hence *we wonder how a constraint added to a design process can contribute to the discovery (and use) of new knowledge independent of Kini in the design process.*

We consider that a constraint added to a design process corresponds to adding P^* to a concept C_0 = XP. The generativity of constraint P^* in C-K theory follows from the definitions given above. Given initial knowledge K_{ini} (that contains knowledge of X, $K_{ini}(X)$, and knowledge of P, $K_{ini}(P)$), a constraint P^* is generative iff XPP^* leads to a conjunction (i.e., XP with additional $P_1 \ldots P_n$ becomes true), and this conjunction is related to P^*. Based on the above presentation of C-K theory, it is clear that a constraint can lead to restrictive generativity (ie based on dependent K) or expansive generativity (I based on independent K). We are interested in the latter case[3].

This calls for examples:

- One example is the design of a safer car (C_0: X = car, P = safer) that anticipates future CO_2 emission regulations (P^*). In designing the safer car, designers might develop original engine solutions that emit very little CO_2.
- On this same C_0, one can consider other types of constraint, such as a safer car "with biomimicry", a safer car "at very low cost", and a safer car "without wheels".
- Another (famous) example is "a new original novel, written without the letter e", which led Georges Perec to write *La Disparition* (1969). This constraint is similar to the famous example of "Green Eggs and Ham", a story written with only 50 different words, studied by Haught-Tromp (2017).

We now analyze the formal conditions for P^* having a generative effect.

1) Formally speaking, adding a constraint P^* to C_0 corresponds to three operations in C-K theory (see Figure 1).

---

[3] Note that for the above model, a conjunction related to P^* is of the form $XPP_1 \ldots P_n$, where P^* is *not* necessarily included. For instance, the task of designing "a safer car without wheels" leads to learning how wheels are related to safety and helps in the design of safer wheels for cars. Formally, it is possible that XPP^* is formulated during the design process, leading to new knowledge that is then used to generate alternatives on XP *without* P^*. This is the so-called "crazy concept" effect.

a) A knowledge injection effect: Any property in C has to be associated with a proposition in K so that adding P^* to XP means that $K(P^*)$ is added to K[4]. $K(P^*)$ is denoted δK_{P^*} (i.e., new knowledge associated with P^*).

b) A partition effect: P^* is added to XP to create a new concept XPP^*.[5]

c) Knowledge discovery driven by XPP^*, which is a learning effect δK stimulated by XPP^*, denoted δK_{XPP^*}: XPP^* might lead to K-expansion (i.e., new learning stimulated by XPP^*), which is different from the knowledge injection effect.

2) Generativity of P^* on XP can come from the above three processes (see Figure 2 and Table 1 below). We analyze each of the processes individually.

a) A constraint can be generative (or not) because the constraint terms themselves bring new knowledge (or not). More formally: knowledge injection $K(P^*) = \delta K_{P^*}$ can be limited and included in $K_{ini}(X)$, in which case it does not relate to generativity (see the above example of wheels) and inversely, δK_{P^*} can also relate to (a lot of) new knowledge, (some of which is) outside of $K_{ini}(X)$, in which case it is more related to generativity. In the example of "a safer car with biomimicry", P^* 'biomimicry' provides much original knowledge, not always related to a "car" or "safety", just because the designer can suddenly consider new knowledge around biology and life science.

---

[4] At least, P^* has to be interpretable in K, meaning that $K(P^*) = \{P^*\}$ at least but that $K(P^*)$ can be broader.

[5] The partition effect is not systematic. It is always possible that P^* finishes the design if XPP^* appears as true or false in K (i.e., P^* provokes a conjunction). This might correspond to the case that the constraint is so strong that the answer is immediately "false"; e.g., the constraint of "a safer car right now". Yet, in the remainder of the paper, we suppose that P^* does not immediately lead to a conjunction and hence the partition effect.

This operation illustrates the complexity of constraint generativity. Knowledge injection independent of $K_{ini}(X)$" is more readily related to (expansive) generativity than knowledge injection dependent on $K_{ini}(X)$ (deductible from $K_{ini}(X)$). However, this condition is insufficient because some elements of δK_{P*} might still be related to $K_{ini}(X)$ (e.g., a car can have aspects of biomimicry, such as in the case that K_{ini} contains "some cars follow aerodynamics laws"). This corresponds to the fact that against an a priori belief, P^* = biomimicry is not necessarily an expansive partition and knowledge discovery driven by XPP^* can only lead to knowledge in $K_{ini}(X)$.

b- A constraint can be generative because it transforms the design task and orients the design exploration in a certain direction. More formally: the partition with P^* (design X with P and P*) can correspond to an expansive partition and hence clearly be outside $K_{ini}(X)$ (e.g., "all cars that we know have wheels" → P^* = "without wheels"). In this case, the constraint P^*, "without wheels", encourages exploration that is "out of the box" and might lead to (expansive) generativity. P^* can also correspond to a restrictive partition and hence clearly be inside $K_{ini}(X)$ (e.g., "all cars that we know have comfortable seats" → P^* = "with comfortable seats").

An expansive partition is more readily related than a restrictive partition to generativity. However, this condition is not sufficient. An expansive partition is insufficient for generativity because it does not necessarily lead to new knowledge contributing to a conjunction. Consider "a safer car (XP) without wheels (P^*)"The designer might simply draw a flying carpet, which can be considered a known imaginary of car mobility and hence already present in $K_{ini}(X)$ (Hooge and Le Du 2016). This proposition would not create new knowledge independent of K_{ini}.

c- The new ‘constrained’ design task will lead to produce new knowledge that can be original, independent of the initial knowledge of th designer. Formally: knowledge discovery driven by XPP^*: δK_{XPP^*} is new knowledge discovered thanks to XPP^* and it can be limited and included in $K_{ini}(X)$, in which case it does not relate to (expansive) generativity (see the above examples of “wheels” and “biomimicry”) or it can relate to (a lot of) new knowledge, (some of which is) outside of $K_{ini}(X)$, in which case it is more related to generativity. In the example of “a safer car, at very low cost”, P^* requiring a very low cost encourages us to explore the costs of a safer car and methods of reducing the costs beyond classical cost saving measures (e.g., second-hand renovation).

This operation also illustrates the complexity of constraint generativity. Knowledge discovery independent of $K_{ini}(X)$” is more readily related to (expansive) generativity than knowledge discovery related to $K_{ini}(X)$. However, this condition is insufficient because some elements of δK_{XPP^*} might still be related to $K_{ini}(X)$; e.g., “a safer car at very low cost” could relate to production in a country having low labor costs. This corresponds to the fact that against an a priori belief, P^* = “at very low cost” is not necessarily an expansive partition, and knowledge injection and knowledge discovery driven by XPP^*might only lead to knowledge in $K_{ini}(X)$.

Hence each elementary process (K-injection, partition, K-discovery driven by XPP*) can both lead to expansive or restrictive generativity. This can be summarized in the figure 2 and the table below.

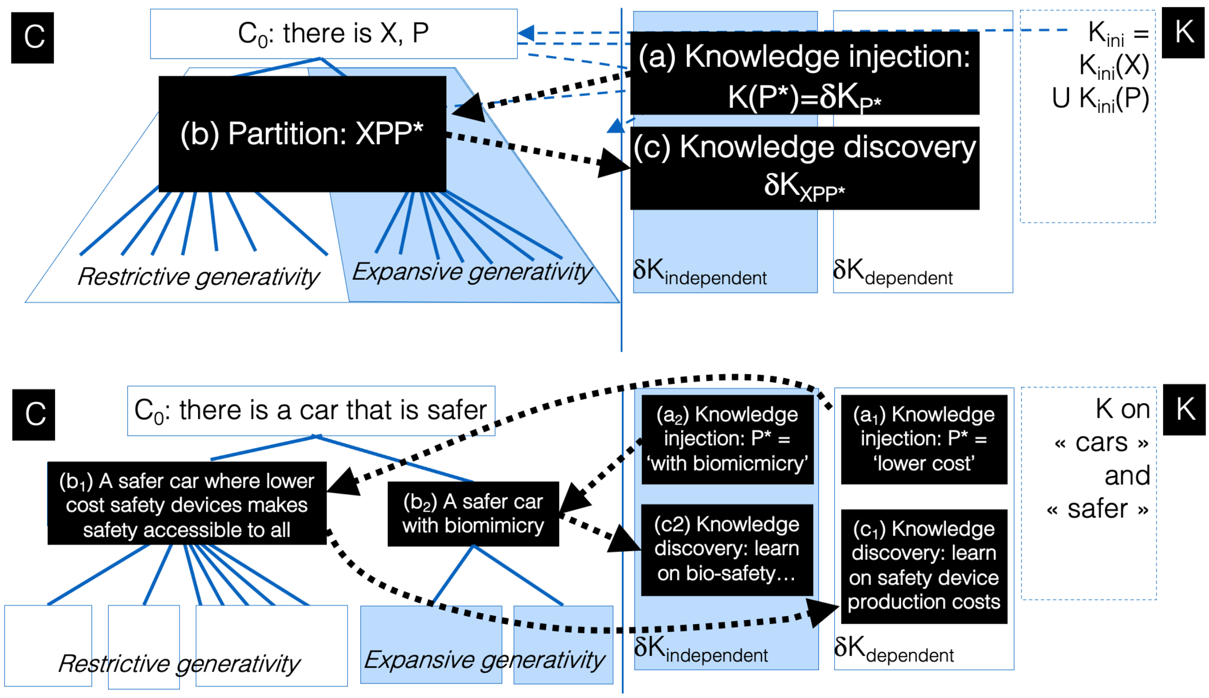

Figure 2: The three effects of a constraint P* on a design process, represented with C-K design theory (figure 2-a), and illustrated by an example (figure 2-b). Figure 2-a: Adding a constraint P^* corresponds to a) injecting knowledge $K(P^*) = \delta K_{P^*}$ - and this new knowledge can be either dependent (not generative) or independent (potentially generative), b) partitioning $C_0 = XP$ with P^*- this partition can be restrictive (not generative) or expansive (potentially generative), and c) creating new knowledge δK_{XPP^*} stimulated by XPP^* - and this new knowledge can be either dependent (non-generative) or independent (potentially generative). Figure 2-b: the design task being "a safer car", the figure illustrates the effect of the constraint P*1 = "with lower cost" with knowledge injection (a1), restrictive partition (b1) and induced knowledge discovery (c1), these three effects resulting here in restrictive generativity; the figure also illustrates the effect of the constraint P*2 = "with biomimicry" with its knowledge injection effect (a2), its expansive partition effect (b2) and induced knowledge discovery (c2), resulting here in expansive generativity.

Operations associated to adding a constraint P* in a C-K design process	Relates to (expansive) generativity if…	Doesn't relate to (expansive) generativity if…
a) Knowledge injection $K(P^*)=\delta K_{P^*}$	If $K(P^*)=\delta K_{P^*}$ is independent of K_{ini}	If $K(P^*)=\delta K_{P^*}$ is dependent of K_{ini}
b) Partition by P*	If it is an expansive partition	If it is a restrictive partition
c) Knowledge discovery, driven by XPP*, δK_{XPP^*}	If δK_{XPP^*} is not empty and is independent of K_{ini}	If δK_{XPP^*} is empty or is depedent of K_{ini}

Table 1: C-K operations associated with a constraint P^* and their potential generativity

II.3. A Model That Contributes To Solving The Paradox Of Generative Constraints

1) On the basis of the above model, the addition of a design constraint P^* to a design task XP corresponds to three operations (i.e., partitioning, knowledge injection, and knowledge discovery) that each contribute (or not) to design generativity; i.e., they contribute to δK for the proof of XP. We can characterize the *necessary condition for (expansive) generativity* and (*at least) one sufficient condition* for generativity.

- The necessary condition for (expansive) generativity is that one of the three operations (or a combination thereof) leads to δK independent of K_{ini} in the sense that it cannot be deduced from/correlated to K_{ini}) and is useful for a conjunction on XP.
- One sufficient condition is that the constraint creates an expansive partition, and this expansive partition leads to δK_{XPP^*} independent of K_{ini} and then to a conjunction.

2) The model sheds light on several paradoxical properties of the effect of constraints on generativity.

a) The model accounts for the aporia of a constraint—seen as a restriction—and (expansive) generativity. On the one hand, in the C-K model, the constraint P^*

restrains the exploration to XPP^* in C, and on the second hand, the C-K model also accounts for other related effects.

- The constraint can open up new areas of knowledge δK_{P*} and δK_{XPP*} outside of K_{ini}.
- The new knowledge can open the door to new exploration $XPP_1 \ldots P_n$, not necessarily under XPP^* (e.g., in the case of creating crazy concepts).
- When the constraint P^* added to XP creates an *expansive* partition, it restrains the exploration to a certain direction, but this direction involves adding a surprising property (P^* outside of K_{ini}). The exploration is thus oriented in a specific direction that is original and new.

b) The model accounts for the fact that a constraint is not systematically generative. This depends on how the three effects (of the partition, δK_{P*}, and δK_{XPP*}) are related to generativity (see Table 1), which means that it depends on the relationship between P^*, K_{ini}, and XP. Let's give some quick illustrations (more detailed cases in the following part):

- Simple case, illustrating "sufficient condition": a constraint provokes an expansive partition (e.g.: "ultra lost cost", text without "e",…) that leads to discover new K ("ultra los cost" leads to new business models, new usages, new product/service architecture...; "text without 'e'" leads to unusual words, new sentences…). Still the constraint can be "too strong" and leads to negative conjunction ("reduce cost of 90% is considered impossible", write a text without vowels is impossible…) and hence no generativity.
- A constraint brings new knowledge ("using knowledge from zoology" often plays this role) and this knowledge, if used, can lead to expansive partitions

and then to original results. This is how “biomimicry” appears as a generative constraint (see below). Still there are counterexamples: this apparently new, independent knowledge is finally restricted to dependent knowledge, eg: my car is a new Beetle, hence it already uses knowledge from zoology!

- Note another interesting case that can be interpreted in this model, the so-called “crazy concept” effect : in a design process under constraint P^*, XPP^* can be “crazy” in the sense that it is inspiring but doesn’t lead to any conjunction yet – Hence XPP* is an expansive partition and this partition provokes knowledge discovery δK_{XPP*} that is independent of K_{ini}. But this new knowledge doesn’t lead to a conjunction “under” XPP^*. At first this configuration doesn’t lead to an expansive generativity. However this new independent knowledge is also now available to provoke other, alternative expansive partitions not under XPP* but still under XP and this new expansive partitions can lead to a conjunction and hence expansive generativity. One example: “XP= a safer car; P*= at no cost” could lead to investigate alternative business models (with insurance companies, with employers, with cities,…) that won’t make possible a safer car ‘at no cost’ but will make possible surprising types of ‘safer car’ (with new insurance contracts, or with new urban development and planning…). It shows that the critical issue is in the discovery of independent knowledge.

III. Interpreting Experiments With A C-K-Based Model Of Constraint Generativity

We now illustrate how the model of a generative constraint corresponds to empirical experiments and how it helps characterize the management of constraints to control generative effects, ie both to achieve generativity or conversely, occasionally, to intentionally avoid generativity.

We review three series of (published) experiments conducted in different contexts, namely laboratory experiments on ideation and defixating leadership, in silico experiments on evolutionary algorithms (EAs), and real-life experiments on biomimicry.

III.1. Constraints In Creativity Cognition: Experiment On Defixation

Fixation is recognized as a critical cognitive phenomenon that impedes generativity (Finke et al. 1992; Ward et al. 1999; Crilly 2015). A series of works have contributed to identifying ways to overcome fixation by adding a specific instruction or constraint in the ideation task (Agogué et al. 2014a; Agogué et al. 2015; Agogué et al. 2014b; Camarda et al. 2021; Ezzat et al. 2018; Ezzat et al. 2017a). We review two series of experiments to show how their results correspond to the prediction obtained with the C-K based model of generative constraint. In each case experiments are done as follows: a "leader" gives to *n* ideators working separately the design task: "propose as many original solutions to make it happen that a hen's egg launched from a height of 10 meters does not break" and this common instruction can be modified with specific additional constraint, as detailed below (with n_i individual in each group i with specific instruction)(detailed results can be found in the publications mentioned below) – based on a reference, it is possible to identify the fixation area associated to the design task; the ideation process results in ideas proposed by the n_i

individuals of the group i with instruction i, the distribution of ideas is analyzed to get a score of originality that is associated to the number of ideas that are considered out of the fixation area. Hence an experimental link between the constraint and the ideas out of th fixation areas.

The correspondence between this series of experiment and the model is as follows: the use of design theory in the field of cognitive psychology (Agogué et al. 2014a; Camarda et al. 2021; Ezzat et al. 2017a) has led to the distinguishing (see Figures 3 and 4) of knowledge that is "easily accessible" ($K_{fixation}$, considered as equivalent to K that is dependent of K_{ini}, non-shaded area in K) and leads to ideas in fixation (non-shaded area in C) and knowledge that is "less accessible" ($K_{defixation}$, considered as independent of K_{ini}, shaded area in K) and is associated with δK and leads to ideas in defixation (shaded area in C). The initial concept is defined by reference to initial knowledge K_{ini}.

We analyze the correspondence between experimental results and the prediction coming from the model of generative constraint:

- Experiments with defixating examples: three groups are made. In group 1, the instruction is "propose as amny original solutions to make it happen that a hen's egg launched from a height of 10 meters does not break"; in group 2, this instruction is completed with "as original as tame an eagle to catch the egg during its fall" ; in group 3, the initial instruction is completed with "as original as using a parachute to slow the fall of the egg" (Agogué et al. 2014a). Results are: group 2 is significantly more original than group 1 that is significantly more original than group 3.

  One can interpret this experiment with our 'constraint' framework: the constraint is either "as original a using a parachute" or "as original as taming an eagle". In the 'parachute' case, knowledge injection δK_{P*} is in the dependent area, XPP^* is not expansive and knowledge discovery δKXP_{P*} (learning driven by XPP^*) is also in dependent area. Therefore, the model predicts no expansive generativity, and this corresponds to the experimental model. In the "eagle" case, knowledge injection δK_{P*} is in the independent area, hence XPP* is an expansive partition and knowledge discovery δK_{P*} driven by XPP* is (very likely) in independent area. Hence the model predicts expansive generativity (see table 2 below), and this corresponds to the experimental model.

Hence these experiments confirm the necessary and sufficient conditions: the 'parachute' case confirms necessary condition (by negation); the 'eagle' case confirms the sufficient condition.

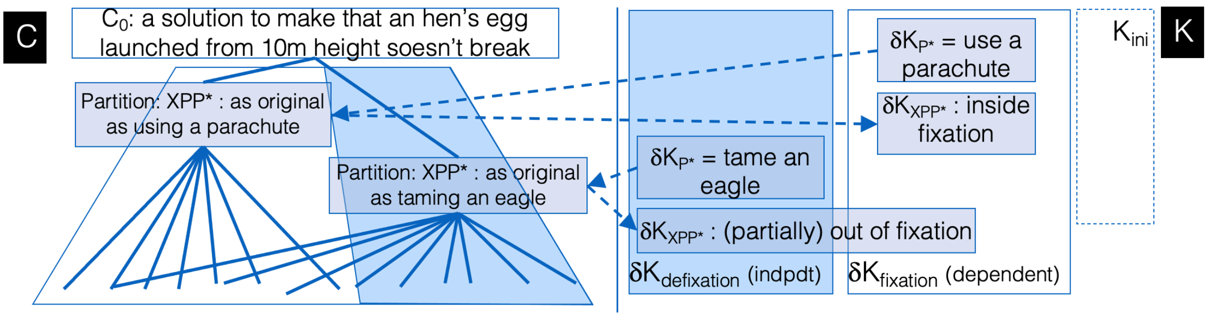

Figure 3: Fixation/defixation effects on idea generation. Example of a parachute: δK_{P*} is in $K_{fixation}$, the partition XPP* is not expansive, δK_{XPP*} is in $K_{fixation}$, Example of taming an eagle: δK_{P*} is in $K_{defixation}$, XPP* is an expansive partition, δK_{XPP*} can be in $K_{defixation}$ (and also in $K_{fixation}$).

Operations associated to adding a constraint P*	Constraint = « as original as using a parachute »	Constraint = « as original as taming an eagle »
a) Knowledge injection $K(P^*)=\delta K_{P*}$	K('parachute')=δK_{P*} is dependent of K_{ini}	K('taming an eagle')=δK_{P*} is independent of K_{ini}
b) Partition by P*	'as original as using a parachute ' is a restrictive partition	'as original as taming an eagle' is an expansive partition
c) Knowledge discovery, driven by XPP*, δK_{XPP*}	P* leads to knowledge dependent (easily associated to) of K_{ini}	P* leads to knowledge independent (not easily associated to) of K_{ini}

Table 2: generative and non-generative constraints in creative cognition experiments with examples

- Experiments with negations: It is known (from previous experiments) that the instruction “make it happen that a hen’s egg launched from a height of 10 meters does not break” leads to fixations in three categories, namely protecting the egg, dampening the shock, and slowing the fall (e.g., with a parachute). Three groups are made. In group 1 (reference), the instruction is “propose as many original solutions to make it happen that a hen’s egg launched from a height of 10 meters does not break”; in group 2, this instruction is completed with “without a parachute” (i.e., the negation of an example); in group 3, the initial instruction is completed with “without slowing the fall, nor damping the shock, nor protecting the egg” (i.e., the negation of an fixation categories). Results are: group 3 (category negation) is significantly more original than group 2 (example negation) and the group 1 (reference) and there is no difference between these two latter groups (Ezzat et al. 2018).

  One can interpret this experiment with our ‘constraint’ framework (see Figure 4 and table 3): the “example negation” constraint does not provide defixating knowledge and is not an expansive partition (because there are still many solutions that are “not a parachute” but are still in the fixation) hence the model predicts that it won’t be generative – this is in line with the empirical result; the “category negation” constraint is an expansive partition (the constraint blocks every (known) fixation path) and leads to generativity by pushing ideators to (re)discover knowledge outside their fixation, hence the model predicts that the experiment will lead to expansive generativity and this corresponds to empirical results. *This experiment illustrates the sufficient condition:* an expansive partition that leads to δK outside of K_{ini}.

Note that δK_{P*} does not come from defixation knowledge (contrary to the 'eagle' example above). This implies that the constraint setter (a sort of constraint manager) does not need to dispose of defixation knowledge, which makes a critical difference with the example of the eagle.

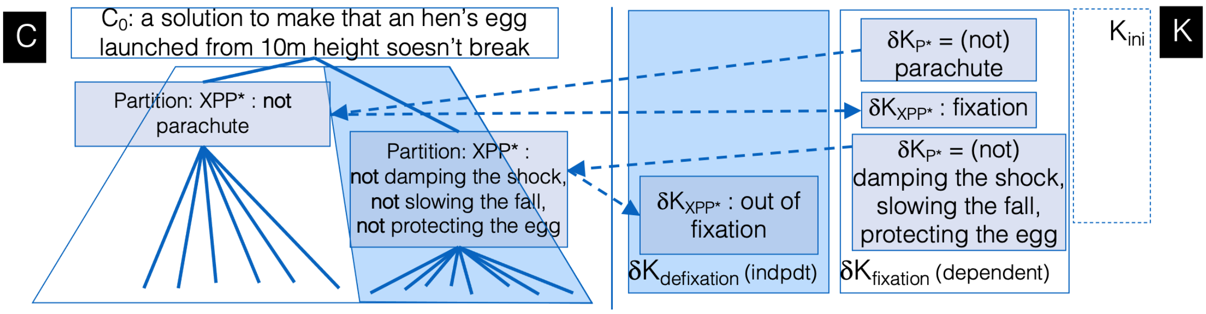

Figure 4: Effect of a negation constraint on idea generation. The negation of an example leads to staying in $K_{fixation}$. The negation of all fixation categories leads to an expansive partition and knowledge discovery in $K_{defixation}$.

Operations associated to adding a constraint P*	Constraint = « Not parachute »	Constraint = « Not damping the shock, not slowing the fall, not protecting the egg
a) Knowledge injection $K(P^*)=\delta K_{P*}$	K('parachute')=δK_{P*} is dependent of K_{ini}	K('damping the shock, slowing the fall, protecting the gg') is dependent of K_{ini}
b) Partition by P*	'not parachute' is a restrictive partition	'not damping the shock, not slowing the fall, not protecting the egg' = expansive partition
c) Knowledge discovery, driven by XPP*, δK_{XPP*}	leads to dependent K (easily associated to K_{ini})	leads to independent K (not easily associated to K_{ini})

Table 3: generative and non-generative constraints in creative cognition experiments with negation.

III.2. Constraints That Make Design Software More Generative

There has long been the logic of a constraint in computer science, where constraints are used for their restrictive role to support convergence toward an acceptable objective or to select among several possible solutions (e.g., in ridge regression). Several algorithms work as heuristics, where constraints allow a satisfactory optimum to be found. A famous example is the evolutionary algorithms (EA) that provide a solution to problems without continuity and hence without gradients (Deb et al. 2002) by "constraining" a trial-and-error process by selecting "elite" solutions and organizing mutation and hybridization among the selected elites. Recently, in this field of the EA (which is also considered an example of so-called generative algorithms (Byrne et al. 2014; Caetano et al. 2020; Hatchuel et al. 2021b; Mountstephens and Teo 2020)), new algorithms that provide greater generativity (i.e., more varied and higher optima) have been proposed; in particular, the novelty search with local competition (NSLC) (Lehman and Stanley 2008; Pugh et al. 2016; Lehman and Stanley 2011) and map elites (ME) (Cully et al. 2015; Fioravanzo and Iacca 2019; Mouret and Clune 2015).

We now analyze how the generativity performances of these algorithms, NSLC and ME, can be explained by the necessary condition and sufficient condition that we identified with a C-K based model of constraint generativity.

The adoption of design theory in the field of EAs (Hatchuel et al. 2021a) leads us to consider K_{ini} = {genes that can combine into specific phenotype ϕ_1} and C_0 = "find a genotype that optimizes ϕ_1 (X=genotype; P=max ϕ_1)". In $K_{dependent}$, we have knowledge that is accessible using the classical EA (hence $K_{dependent}$ corresponds to $K_{with\ classical\ EA}$ - non-shaded area in K)) and leads to genotypes with optimal ϕ_1 obtained with the classical EA (non-shaded area in C) and we have $K_{independent}$ that is knowledge beyond this knowledge accessible with the classical EA (hence $K_{independent}$ corresponds to $K_{beyond\ classical\ EA}$ - shaded area in K) and that leads to genotypes with new, possibly better optima for ϕ_1 (shaded area in C).

- A novelty search with the local competition (NSLC) EA implies that the algorithm keeps interesting genotypes if their phenotype varies one from the other (i.e., the novelty criterion, which differs from the usual ϕ_1 optimizing criterion and is not present initially in the classical EA algorithm). This novelty criterion leads to the investigation of original paths in C and hence the formulation of the genotypes already obtained with the classical EA as well as new genotypes that are also 'acceptable' in ϕ_1 optima (local optimum) but with original ϕ_2. Even more: actually ϕ_1-optima obtained with EA corresponded to specific ϕ_1 values, and for these specific ϕ_2 values, NSLC will finally discover genotypes with *higher* ϕ_1. Hence NSLC leads to expansive generativity by comparison with classical EA.

- The ME EA also introduces an alternate ϕ_2 phenotypical criterion (not present for the classical EA, such that δK_{P*} is outside $K_{with\_classical\_EA}$) and organizes the systematic exploration by selecting ϕ_1 elites for each specific niche defined by ϕ_2 intervals. The ME EA *systematically* explores genotypes along ϕ_2 and thus, in C, XPP* opens a range of individuals (one so-called elite individual with phenotype (ϕ_1, ϕ_2) for each niche defined by one interval along ϕ_2) and in K it corresponds to the exploration of more genotypes beyond the classical EA (δK_{XPP*}). Just like NSLC, ME will result in genotypes already obtained with the classical EA as well as new genotypes that are also 'acceptable' in ϕ_1 optima (local optimum) but with original ϕ_2. and ME will also results in better ϕ_1-optima in the ϕ_2-regions already explored by classical EA. The exploration along ϕ_2 is more systematic with ME and this results in more expansive partitions: new solutions in new ϕ_2-regions as well as better optima in known ϕ_2-regions.

ME and NSLC EA thus create more varied and more ϕ_1-optimal individuals. This increase in generativity corresponds to three factors (see table 4 below): knowledge injection outside of $K_{with\_classical\_EA}$ (δK_{P*} provided by the ϕ_2 parameter), the possibility of expansive partitions (for certain values of ϕ_2 not already explored by the classical EA – this process is reinforced in ME that systematically explore ϕ_2), and knowledge discovery (δK_{XPP*} guided by ϕ_2). ME and NSLC EA illustrate a sufficient condition: if ϕ_2 is well chosen, then it is possible that certain ϕ_2 intervals correspond to expansive partitions and hence to δK_{XPP*} outside of $K_{with\_classical\_EA}$.

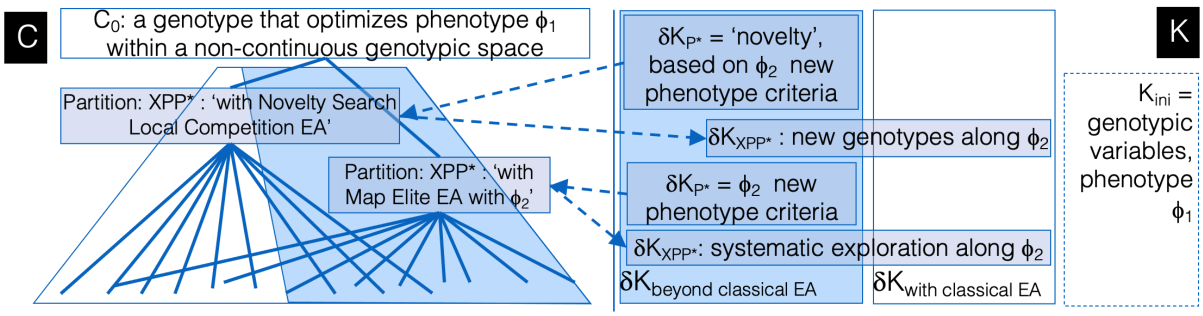

Figure 5: Generativity with NSLC and ME EAs. ME and NSLC EA introduce new phenotypical criteria ϕ_2 in K. These new phenotypical criteria increase the chance of provoking an expansive partition in K and thus the chance of discovering new knowledge. The range of ϕ_2 is explored in a more systematic way in ME than in NSLC, resulting in more expansive generativity.

Operations associated to adding a constraint P*	Constraint = « with Novelty Search Local Competition EA»	Constraint = « with Map Elite EA»
a) Knowledge injection $K(P^*)=\delta K_{P^*}$	NSLC uses new phenotypic variable $\phi_2 = \delta K_{P^*}$ independent of K_{ini}	ME uses new phenotypic variable $\phi_2 = \delta K_{P^*}$ independent of K_{ini}
b) Partition by P*	'with NSLC' = explore variety of solutions along ϕ_2 (can be expansive)	'with ME' = explore systematically solutions along ϕ_2 (is expansive)
c) Knowledge discovery, driven by XPP*, δK_{XPP^*}	P* can lead to genotypes beyond classical EA	P* can lead to genotypes beyond classical EA

Table 4: constraints that make evolutionary algorithms more generative.

III.3. Biomimicry As A Generative Constraint

Several empirical studies have been conducted to analyze and experimentally determine mechanisms through which biomimicry can increase generativity. Some of these studies were based on C-K theory (Freitas Salgueiredo and Hatchuel 2016; Nagel et al. 2016; Pidaparti et al. 2020; Prabaharan et al. 2019). Freitas Salgueiredo & Hatchuel (2016) showed that biomimicry stimulates creativity while adding very little knowledge (δK_{P*} = 'biomimicry' ≅ ∅) but creating an expansive partition in C (XP "with biomimicry" adds an unusual partitioning attribute to XP), which might lead to δK_{XPP*} outside of knowledge usually used to solve XP (hence independent). The authors showed that the critical operation is in δK_{XPP*}. Knowledge discovery under "biomimicry" could provide original, relevant knowledge outside of knowledge usually used by designers (independent). However, it could also remain sterile, without bringing any relevant bio-related knowledge or only finding bio-related knowledge that was actually already related to K_{ini} (dependent). This confirms that *biomimicry can meet the necessary condition of generative constraint iff knowledge discovery* δK_{XPP*} *uncovers new knowledge outside of* K_{ini} (see table 5 below)

On the basis of this result, Nagel et al (Nagel et al. 2016; Pidaparti et al. 2020; Prabaharan et al. 2019) experimented on many ways to manage this critical δK_{XPP*}, based on C-K patterns and clear access to bio-knowledge during the design process. The results of the experiments confirm that a good management of δK_{XPP*} is critical for making biomimicry act as a generative constraint.

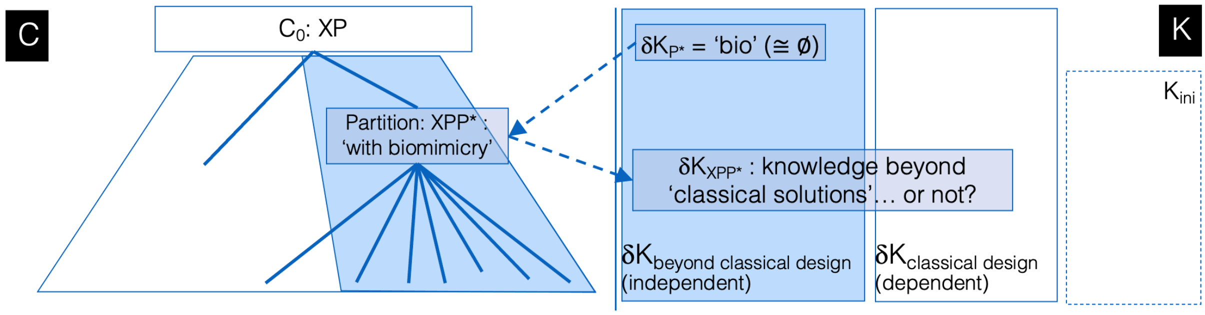

Figure 5: Biomimicry as a generative constraint. δK_{P*} is outside classical knowledge (hence independent) but it might be quasi-empty. P^* = "with biomimicry" is an expansive partition. However, it is uncertain whether δK_{XPP*} is outside classical knowledge (independent).

Operations associated to adding a constraint P*	Constraint = « with biomimicry »
a) Knowledge injection $K(P^*)=\delta K_{P*}$	δK_{P*} is independent of K_{ini} but not very rich (designers usually have limited knowledge on bio)
b) Partition by P*	Expansive partition
c) Knowledge discovery, driven by XPP*, δK_{XPP*}	Tends to leads to iindependent K (although dependent is also possible)

Table 4: how biomimicry can be interpreted as a generative constraint.

IV. Solving The Paradox Of Generative Constraints—Managerial Implications

IV.1. Contribution: Conditions Of A Generative Constraint

On the basis of design theory and with confirmation in a series of experiments, a design constraint P^* added to a design task "there is an X with property P" (initially having an unknown relation to knowledge on X, $K_{ini}(X)$, and knowledge on P, $K_{ini}(P)$) corresponds to three operations (i.e., partitioning, knowledge injection, and knowledge discovery) that can each contribute (or not contribute) to design generativity. Partitioning can be expansive (or restrictive), knowledge injection and knowledge discovery can be outside of (independent) knowledge related to $K_{ini}(X)$ and knowledge related to P $K_{ini}(P)$ (or correlated to $K_{ini}(X)$ or $K_{ini}(P)$).

A necessary condition for a generative constraint is that independent knowledge is produced in the design process "under constraint" and the model shows and the experiments confirm that this can happen through one of the three mechanisms – or said formally: one of the three operations leads to δK independent of $K_{ini}(X)$ and K(P) and leads to a conjunction. A sufficient condition for the generative constraint is that the constraint creates an expansive partition and this expansive partition leads to knowledge discovery δK_{XPP*} independent of $K_{ini}(X)$ and $K_{ini}(P)$. Both conditions have been confirmed by empirical studies in very different contexts (lab experiment on defixating leadership, in silico experiments on generative algorithms, real life experiment on b.

The above result reinforces results on constraints and creativity mentioned in the introduction.

- Frequent inconclusiveness? The model explains that a given constraint type can be either generative or restrictive depending on whether the constraint P* inject independent knowledge δK_{P*}, provokes expansive partition and leads to knowledge discovery δK_{XPP*}. independent of K_{ini}.
- Plausibility of a U-shaped relation? In specific cases, the conditions result in an inverted U-shaped relation (eg cost: low cost / strongly lower cost / no cost: this constraint will correspond respectively to restrictive partition / expansive partition / negative conjunction). However, the above result demonstrates that the relation depends also on K-discovery, which can explain a deviance from U-shape relation: "lower cost" constrain can become generative it leads to discover independent

knowledge; respectively “no cost” can also become generative if it leads to independent knowledge.

- Constraint aporia? The constraint restricts the exploration (XPP^*), but this restriction can be associated with many openings in the design process: a) knowledge injection and discovery can open up new knowledge, b) this new knowledge can open up new explorations in the C-space outside the “restriction” imposed by the constraint P^*, and, c) last but not least, the restriction itself (XPP^*) being an expansive partition imposes an *unexpected* property on X, an unexpected property that is, in itself, an opening..

IV.2. Implications For Managing Constraints For Generativity

The above results have multiple implications in management and education. As seen in empirical cases, a better understanding of the generative constraint can support defixating leadership, help develop efficient biomimicry-driven design processes, and even help identify relevant algorithms for generative design tools.

More deeply, the results correspond to managerial challenges.

1) First: managing knowledge injection by the constraint. Managing the generative impact of a constraint consists in evaluating how this constraint injects *independent* knowledge. There are cases where teachers or managers know the team well enough to anticipate the dependent knowledge associated to a certain design task – in this case it might be relatively easy to identify constraints that will bring independent knowledge – just as biomimicry does for engineering students. Still in some cases, the team is too large or too unknown, or the design task itself is too surprising and the professor / the manager is not able to predict what is ‘dependent’ knowledge for the team. This will then require complex investigations, (organizational) learning processes, and organizational procedures. More

research would be needed to explore in more depth the various ways and means to accomplish this management action.

2) Second: managing knowledge discovery induced by the design exploration under the constraint. Managing the *knowledge creation* δK_{XPP*} process stimulated by the constraint consists in producing independent knowledge inspired by the chimera XPP*. For managers or for teachers, it consists in supporting the learning / exploratory processes induced by a constraint, hence providing ways an means to learn/produce knowledge in unexpected directions – eg. mechanical engineering students might begin to learn on biology. This might sound simple in experimental situations, still it appears that in real life situations, this type of learning is all the more complex that designers tend to rely on learning devices that are related to what they already know (same discipline, same skills,…) whereas the managerial recommendation exactly means to learn in new independent areas (new devices, new experts, new networks…). Research works have shown that some software devices might enhance this process learning independent knowledge (e.g., Arrighi et al., 2015)) or this learning process would require specific organizational processes to help actors quickly absorb external knowledge or create it (Klasing Chen 2015; Plantec et al. 2021).

3) The above mentioned managerial recommendations are not easily realized – this is probably not surprising when one remembers the large inconclusiveness on the management of generative constraint: the model finally predicts that efficient management of generative constraint requires strong insights on available knowledge and the independent knowledge to be acquired/created. This would be all the more difficult in complex, multi-disciplinary teams where available knowledge and missing, independent knowledge are not easily identifiable. Yet the value of the recommendation to take care of 'independent knowledge' associated to a constraint might precisely be in the fact that it is not self-evident and intuitive and call for

further research to be more easily implemented in management procedures and managerial behavior.

IV.3. Discussion: Generalizing The Logic Of Generative Constraints Beyond The Design Task

This chapter addresses the generative constraint in the case of the so-called task constraint. How can one extend the results to a so-called situation constraint (i.e., a constraint relating to a design situation), such as procedural constraint, team constraint, or organizational constraint?

Even if the model is described from the perspective of P^* being a "task variable" (Haught-Tromp, 2017), the model can be generalized with P^* relating to a so-called *situation* constraint (or also contextual constraints such as budget, time, organization, etc…), as long as the situation constraint can be translated into a P^* added to the design task.

We give the following examples.

- A design process with a procedural constraint "rely on user involvement" readily leads to P^* = "relying on user involvement".
- A design process that follows the set-based design process, in which several solutions have to be designed at each stage, can easily be translated into P^* = "design always several solutions at each stage" (see (Sobek 1996; Sobek et al. 1999)).
- A design process where the design team comprises only women (a real (and excellent) case study of Volvo by Backman & Börjesson (Backman and Börjesson 2006)) can easily be translated into P^* = "designed by women".

In these cases, three operations should be considered.

- Does the constraint correspond to an expansive partition? For the example P^* = "with user involvement", in businesses where user involvement is frequent, the answer is no.

In businesses where user involvement is rare, this constraint can be expansive (depending on the knowledge on the user).

- Does the constraint inject new knowledge? In the example P^* = "design at least two alternatives at each stage", the answer is no. Still, this constraint could have generative effects relating to the two other operations.
- Does the constraint lead to the discovery of new knowledge? The example P^* = "designed by women" might or might not lead to the discovery of new knowledge (as studied by Backman et al.).

Self-evidently sufficient and necessary conditions also apply in these cases.

IV.4. Opening: Preservation Constraint And The Design Of Creation Heritage

Are there constraints that inherently kill generativity? Even if it is difficult to address this question in general terms, the question at least leads us to wonder whether generativity can be limited or, conversely, enhanced by a constraint of *preservation*. This can be a constraint of "tradition preservation" but the question can also be more general in these times of environmental transition, where designers are looking for creative sustainable solutions that address grand challenges and innovation is often deeply related to preservation; e.g., preserving natural resources, preserving biodiversity, preserving a way of life, preserving health, preserving wealth, or preserving mobility. In such cases, we are used to considering that the constraint is in direct conflict with innovation and creation in that one can only be achieved at the cost of the other. But is this intuitive answer really correct? What does the model teach us about this?

a) The model clarifies why the "preservation" constraint is difficult to handle. First, knowledge injection is difficult; i.e., do we know what has to be preserved ex ante. Second, it is not easy to see how to deal with the other two operations: partitioning (will "with preservation" act as a restrictive or expansive partition?) and knowledge discovery (what will be discovered thanks to the preservation constraint?).

b) However, these difficulties also present interesting ways to deal with such a constraint. It appears that "with preservation" can precisely lead to surprising "creative partitions" (e.g., "a disruptive solution to be made compatible with preservation"; see Carvajal Pérez et al., 2020; Hatchuel et al., 2019) and interesting knowledge discovery (what has to be preserved will actually be learnt and discovered while walking; see Harlé et al., 2021). We thus see how the model inspires the creation of new ways to deal with constraints.

Acknowledgment

We thank Edanz (www.edanz.com/ac) for editing a draft of this manuscript. We thank Thomas Gillier for his friendly review of this chapter.

References

Acar OA, Tarakci M, van Knippenberg D (2018) Creativity and Innovation Under Constraints: A Cross-Disciplinary Integrative Review. *Journal of Management* 45 (1):96-121.

Agogué M, Kazakçi A, Hatchuel A, Le Masson P, Weil B, Poirel N, Cassotti M (2014a) The impact of type of examples on originality: Explaining fixation and stimulation effects. *Journal of Creative Behavior* 48 (1):1-12.

Agogué M, Le Masson P, Dalmasso C, Houdé O, Cassotti M (2015) Resisting classical solutions: The creative mind of industrial designers and engineers. *Journal of Psychology of the Aesthetics, Creativity and the Arts* 9 (3):313-318.

Agogué M, Poirel N, Pineau A, Houdé O, Cassotti M (2014b) The impact of age and training on creativity: a design-theory approach to study fixation effects. *Thinking Skills and Creativity* 11:33-41.

Amabile TM (1998) How to kill creativity. *Harvard Business Review* september-October 1998:77-87.

Amabile TM, Barsade SG, Mueller JS, staw BM (2005) Affect and Creativity at Work. *Administrative Science Quarterly* 50 (3):367-403.

Arrighi P-A, Le Masson P, Weil B (2015) Addressing constraints creatively: how new design software helps solve the dilemma of originality and feasibility". *Creativity and Innovation Management* 24 (2):247-260.

Arrighi P-A, Le Masson P, Weil B, Kazakçi A (2016) Uncovering the specificities of CAD tools for industrial design with design theory – style models for generic singularity. *International Journal of Design Creativity and Innovation*:1-19.

Backman M, Börjesson S (2006) Vehicles for attention creation: the case of a concept car at Volvo Cars. *European Journal of Innovation Management* forthcoming.

Boden MA (1999) Computer Models of Creativity. In: Sternberg RJ (ed) *Handbook of creativity*. Cambridge University Press, Cambridge, pp 351-372

Byrne J, Cardiff P, Brabazon A, O′Neill M (2014) Evolving parametric aircraft models for design exploration and optimisation. *Neurocomputing* 142:39-47.

Caetano I, Santos L, Leitão A (2020) Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research* 9 (2):287-300.

Camarda A, Bouhours L, Osmont A, Le Masson P, Weil B, Borst G, Cassotti M (2021) Opposite effect of social evaluation on creative idea generation in early and middle adolescents. *Creativity Research Journal* 33 (4):399-410.

Crilly N (2015) Fixation and creativity in concept development: The attitudes and practices of expert designers. *Design Studies* 38:54-91.

Cully A, Clune J, Tarapore D, Mouret J-B (2015) Robots that can adapt like animals. *Nature* 521 (7553):503-507.

Damadzic A, Winchester C, Medeiros KE, Griffith JA (2022) [Re]thinking outside the box: A meta-analysis of constraints and creative performance. *Journal of Organizational Behavior* n/a (n/a).

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6 (2):182-197.

Eckert CM, Stacey M, Wyatt D, Garthwaite P (2012) Change as little as possible: creativity in design by modification. *Journal of Engineering Design* 23 (4):337-360.

Epstein R (1999) Generativity Theory. In: Runco M, Prtzker S (eds) *Encylopedia of Creativity*, vol 1. Academic Press, pp 759-766

Ezzat H, Agogué M, Le Masson P, Weil B, Cassotti M (2018) Specificity and Abstraction of Examples: Opposite Effects on Fixation for Creative Ideation. *The Journal of Creative Behavior* 54 (1).

Ezzat H, Camarda A, Cassotti M, Agogué M, Houdé O, Weil B, Le Masson P (2017a) How minimal executive feedback influences creative idea generation. *PLOS ONE* 12 (6):e0180458.

Ezzat H, Le Masson P, Weil B (2017b) Extending lab results to advices for leadership facilitating creativity in organizations. *CERN IdeaSquare Journal of Experimental Innovation* 1 (2).

Finke RA, Ward TB, Smith SM (1992) *Creative Cognition*. MIT Press, Boston, MA

Fioravanzo S, Iacca G (2019) Evaluating MAP-Elites on constrained optimization problems. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*.

Freitas Salgueiredo C, Hatchuel A (2016) Beyond analogy: A model of bio-inspiration for creative design. *AI EDAM* 30 (Special Issue 02):159-170.

Gillier T, Chaffois C, Belkhouja M, Roth Y, Bayus BL (2018) The effects of task instructions in crowdsourcing innovative ideas. *Technological Forecasting and Social Change*.

Groop J, Ketokivi M, Gupta M, Holmstrom J (2015) Theory of Constraints as Design Science: The Case of Improving Home Care. *Academy of Management Proceedings* 2015 (1).

Hatchuel A, Klasing Chen M (2017) Creativity under strong constraints: the hidden influence of design models. *European Review* 25 (2):194-207.

Hatchuel A, Le Masson P, Reich Y, Subrahmanian E (2018) Design theory: a foundation of a new paradigm for design science and engineering. *Research in Engineering Design* 29:5-21.

Hatchuel A, Le Masson P, Reich Y, Weil B (2011a) A systematic approach of design theories using generativeness and robustness. In: *International Conference on Engineering Design, ICED'11*, Copenhagen, Technical University of Denmark, 2011a. p 12

Hatchuel A, Le Masson P, Thomas M, Weil B (2021a) What is generative in generative design tools? Uncovering topological generativity with a C-K model of evolutionary algorithms. *Proceedings of the Design Society* 1:3419-3430.

Hatchuel A, Le Masson P, Thomas M, Weil B (2021b) What is Generative in Generative Design tools? Uncovering topological generativity with a C-K model of evolutionary algorithms. In: *23rd International Conference on Engineering Design ICED'21*, Gothenburg, Sweden, 2021b. p 10

Hatchuel A, Le Masson P, Weil B (2011b) Teaching Innovative Design Reasoning: How C-K Theory Can Help to Overcome Fixation Effect. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 25 (1):77-92.

Haught-Tromp C (2017) The Green Eggs and Ham hypothesis: How constraints facilitate creativity. *Psychology of Aesthetics, Creativity, and the Arts* 11 (1):10-17.

Hooge S, Le Du L (2016) Collaborative Organizations for Innovation: A Focus on the Management of Sociotechnical Imaginaries to Stimulate Industrial Ecosystems. *Creativity and Innovation Management* 25 (3):311-330.

Johnson-Laird P (1988) Freedom and constraint in creativity. In: Sternberg RJ (ed) *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge University Press, New York, USA, pp 202-2019

Keupp MM, Gassmann O (2013) Resource constraints as triggers of radical innovation: Longitudinal evidence from the manufacturing sector. *Research Policy* 42 (8):1457-1468.

Klasing Chen M (2015) *Smart low cost design in complex value networks : The case of public transport* MINES ParisTech PSL Research University, Paris

Krish S (2011) A practical generative design method. *Computer-Aided Design* 43 (1):88-100.

Lehman J, Stanley K (2011) Evolving a diversity of creatures through novelty search and local competition. In: *Genetic and Evolutionary Computation Conference, GECCO'11*, 2011. pp 211-218. doi:10.1145/2001576.2001606

Lehman J, Stanley KO (2008) Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. Paper presented at the *Artificial Life XI*, Winchester. United Kingdom,

Mountstephens J, Teo J (2020) Progress and Challenges in Generative Product Design: A Review of Systems. *Computers* 9 (4).

Mouret J-B, Clune J (2015) Illuminating search spaces by mapping elites. *ArXiv*.

Nagel JK, Pittman P, Pidaparti R, Rose C, Beverly C (2016) Teaching bioinspired design using C–K theory. *Bioinspired, Biomimetic and Nanobiomaterials* 6 (2):77-86.

Onarheim B (2012) Creativity from constraints in engineering design: lessons learned at Coloplast. *Journal of Engineering Design* 23 (4):323-336.

Onarheim B, Biskjaer MM (2013) An Introduction to 'Creativity Constraints'. In, 2013.

Pidaparti RM, Prabaharan GP, Nagel JKS, Rose CS (2020) Engineering Design Innovation through C-K theory based templates. *Journal of STEM education* 21 (1):58-63.

Plantec Q, Le Masson P, Weil B (2021) Impact of knowledge search practices on the originality of inventions: a study of oil & gas industry through dynamic patent analysis. *Technological Forecasting and Social Change* (accepted, to be published).

Prabaharan GP, Nagel JK, Rose CS, Pidaparti RM (2019) Investigation of C-K Theory Based Approach for Innovative Solutions in Bioinspired Design. *Designs* 3 (3):39.

Pugh JK, Soros LB, Stanley K (2016) Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers Robotics AI* 3:40.

Rosso BD (2014) Creativity and Constraints: Exploring the Role of Constraints in the Creative Processes of Research and Development Teams. *Organization Studies* 35 (4):551-585.

Sobek DK (1996) A set-based model of design. *Mechanical Engineering* 118 (7):78-81.

Sobek DK, Ward AC, Liker JK (1999) Toyota's Principles of Set-Based Concurrent Engineering. *Sloan Management Review* Winter 1999:67-83.

Sternberg RJ, Kaufman JC (2012) Constraints on Creativity: Obvious and Not So Obvious. In: Kaufman JC, Sternberg RJ (eds) *The Cambridge Handbook of Creativity*. Cambridge University Press, New York, pp 467-482. doi:DOI: 10.1017/CBO9780511763205.029

Thomas LDW, Tee R (2021) Generativity: A systematic review and conceptual framework. *International Journal of Management Reviews* n/a (n/a).

Vajna S, Clement S, Jordan A, Bercsey T (2005) The Autogenetic Design Theory: An evolutionary view of the design process. *Journal of Engineering Design* 16 (4):423-440.

Ward TB, Smith SM, Finke RA (1999) Creative Cognition. In: Sternberg RJ (ed) *Handbook of Creativity*. Cambridge University Press, Cambridge, pp 189-212

Zittrain JL (2006) The Generative Internet. *Harvard Law Review* 119:1974-2037.