

# **A search and optimization perspective on conceptual design**

**Ehud Kroll, Gil Weisbrod**

**Faculty of Aerospace Engineering  
Technion–Israel Institute of technology, Haifa, Israel**

## **Abstract**

Conceptual design using the method of Parameter Analysis is related in this paper to the tree structure formed in C-space of C–K Theory. This, in turn, leads to an analogy to the search tree in branch-and-bound optimization. We show that the classical branch-and-bound method can be generalized to include a dynamic and expanding state space, operators as design moves that are not pre-defined, and a cost function that consists of learning-based evaluation of the artifact. The generalized branch-and-bound modeling can be used as a well-understood notional framework to study and compare human design methods and to develop computerized design assistants.

## **Keywords**

*Conceptual design, Branch-and-bound, Parameter analysis*

# 1 INTRODUCTION

Efficient exploration of many alternatives and the quest for the best solution are inseparable from design. We propose here that the branch-and-bound (B&B) method from operations research (OR) can be generalized to provide a conceptual framework to describe design processes. B&B (Hillier and Lieberman, 2005) is a technique for finding optimal solutions to integer programming problems with a very large number of solutions. The basic idea is to divide and conquer so only a small fraction of the feasible solutions needs to be examined. An original large problem is divided (the branching) into smaller and smaller sub-problems that are more manageable. The conquering is done by bounding how good the best solution in the subset of feasible solutions can be, and then discarding the subset if its bound indicates that it cannot possibly contain an optimal solution. B&B has been developed since the early 1960s, representing the simulation of human decision making as rational behavior: choosing an alternative that provides the highest utility, that is, the optimal one.

Around the same time, problem solving as search became a major interest in artificial intelligence (AI). Weak search methods, capable of solving particular kinds of problems, were followed in AI by heuristic paradigms that could explore large solution spaces efficiently; yet, they were applied mostly to “toy problems” (Tong and Sriram, 1992). Simon (1972) introduced the notion of “bounded rationality” to denote the limitations due to incomplete information about the alternatives and their consequences, human cognition, and resources, all leading to making “satisficing” decisions rather than optimizing (Badke-Schaub and Eris, 2014). Simon’s “science of design” (Simon, 1969) thus incorporated his involvement in AI and shifted the focus from decision theory to problem solving and to subjectivity in the sense of using rules of thumb, heuristics (Hatchuel, 2001), in order to make the search for solutions more efficient.

Over the last several decades, AI-based expert systems that incorporate domain knowledge have been developed to automate many design tasks; for example, VLSI (Gries, 2004), architectural (Mitchell, 2001) and mechanical design (Li et al., 2013). An overview of AI applications to design can be found in (Antonsson and Cagan, 2001). Simultaneously, B&B was shown to be related to many search algorithms in AI (Nau et al., 1984) and to notions of breadth-first, depth-first, best-first, etc., which refer to the order of expanding the nodes of the search graph to form subsets of the solution space (Korf, 2010). Cagan et al. (1997) describe an approach and future work needed to integrate AI techniques and optimization (the OR approach). They note that AI techniques are limited in their search for general designs, while few problems can be solved by structured optimization methods alone. Therefore, only the combined AI and OR approach can provide the problem-specific reasoning, symbolic representations, and powerful numerical optimization search that are needed in design.

The AI “design as problem solving” paradigm has prevailed for many years, attempting to approximate design processes with the tree-structured search space and the B&B heuristics used for simulating games or theorem proving. However, Pearl (1984) points to the fact that the emphasis of B&B methods is on the *split-and-prune* paradigm that is effective in establishing completeness and optimality, while the AI approach is concerned with the *generate-and-test* viewpoint, which is more relevant to creating or constructing new objects while searching for solutions. Heuristic search in the context of path-seeking problems has been studied both in OR and AI, with the purpose of increasing efficiency. The most common use of heuristic information has been the bounding functions that control the B&B search, as in AI’s popular heuristic shortest-path algorithm called  $A^*$  (e.g., Russell and Norvig, 1995).

Dym and Brown (2012, Ch. 6) show how AI-based problem-solving methods can be used to model the ways we address design tasks. In particular, they draw an analogy between searching game trees and design, including issues of search directions and strategies, characterizing design as path-finding problems and constraint-satisfaction problems, and the notion of using domain knowledge (in the form of heuristics) to guide the search process by pruning branches of the tree that are unlikely to yield a solution. Together with surveying some of the classical knowledge-based design systems from the 1980s, they also point to some of the difficulties of solving real design problems by AI methods: “...we can model only the simplest design problems as a formal AI search that starts with a given: state space, well-defined goal nodes, start nodes, and operators.”

In contrast to work that attempts to mimic human problem solving by a computer, fewer efforts have been reported on trying the opposite: applying notions of computer algorithms to understand human

reasoning. Interestingly, one of the earliest papers to introduce the modern C–K design theory (Hatchuel and Weil, 2002) contrasts Simon’s notion of bounded rationality with “expandable rationality”—the generation of new concepts and new knowledge—and claims that the B&B approach uses restrictive partitioning of the static search space and thus is unable to describe innovative design processes. To accommodate inventions and surprises, expansive partitioning should be applied by adding attributes to concepts and changing their definition. Perhaps the most notable suggestion made in their paper is to generalize the B&B method in order to enable expandability by allowing solutions that are not fully specified at the outset, expansive partitions that change the definition of concepts, assessment of concepts relative to existing and new knowledge (“K-validation”), and allowing failure of the design process. This work shows how a computer-based optimization scheme was the basis for deriving a design theory that is aimed at explaining human design processes.

More recently, Kazakçi (2014) has argued that describing design by formal AI models hinders the study of creativity, because the traditional methods use a paradigm that he calls “the world as it is”. In contrast, design reasoning follows “the world as it can be” paradigm, for which the “imaginative constructivism” model is proposed. In it, new types of objects are imagined and new methods for building those objects are sought. In a sense, this work attempts to enrich the field of AI with insights from design theory.

We can conclude that design processes in general cannot be treated as a mere search problem because the state space is not known, the goal state is not given, and often even the root state (the task) is ill-defined and evolves together with its solution (Maher and Tang, 2003). In addition, the operators—the actions that transform one state into another—are usually not known a priori. However, this paper suggests that the notions of search algorithms can be very useful in studying and explaining design processes, in general, and how the number of design alternatives to be explored can be reduced before arriving at a satisfactory solution, in particular. Search and design problems also share a common theme of optimization in a broad sense. Design is not optimization in the “classic” computational context, but it is concerned with finding *good* solutions, not just *any* solution. It also tries to reach the solution in an efficient manner, that is, with minimum resources such as time and knowledge acquisition effort. Therefore, to better understand the efficiency aspects of design methods, an optimization framework needs to be consulted.

## **2 THE PARAMETER ANALYSIS METHOD OF CONCEPTUAL DESIGN**

Parameter analysis (Kroll et al., 2001; Kroll, 2013) is an empirically-derived method for conceptual design. It was developed initially as a descriptive model after studying designers in action and observing that their thought process involved continuously alternating between conceptual-level issues (concept space) and descriptions of hardware (configuration space). The result of any design process is certainly a member of configuration space, and so are all the elements of the design artifact that appear, and sometimes also disappear, as the design process unfolds. Movement from one point to another in configuration space represents a change in the evolving design’s physical description, but requires conceptual reasoning, and this is done in concept space. The concept space deals with “parameters”, which in this context are functions, ideas and other conceptual-level issues that provide the basis for anything that happens in configuration space. Moving from concept space to configuration space involves a realization of the idea in a particular hardware representation, and moving back, from configuration to concept space, is an abstraction or generalization, because a specific hardware serves to stimulate a new conceptual thought. As will be shown later, concept space in PA is epistemologically different from C-space in C–K theory.

To facilitate the movement between the two spaces, a prescriptive model was conceived, consisting of three distinct steps. The first step, Parameter Identification (PI), is the recognition of the most dominant issues at any given moment during the design process. These may include the dominant physics governing a problem, a new insight into critical relationships between some characteristics, an analogy that helps shed new light on the design task, or an idea indicating the next best focus of the designer’s attention. The second step is Creative Synthesis (CS), representing the generation of a physical configuration based on the issue recognized within the previous PI step. The configurations allow one to see new key parameters, which will again stimulate a new direction for the process. The third step is Evaluation (E), which facilitates moving away from a physical realization back to parameters or concepts. Evaluation is important because one must consider the degree to which a

specific implementation represents a possible solution to the entire problem. Evaluation also uncovers the weaknesses of the configurations and points out possible areas of improvement for the next design cycle. The unique role played by the evaluation step is elaborated later.

PA's repetitive PI-CS-E cycles are preceded by a Technology Identification (TI) stage of determining the most challenging functional aspect of the task and looking into fundamental technologies and physical principles that can be used to address it. A cursory listing of each candidate technology's pros and cons follows, leading the designer to pick the one that seems most likely to succeed.

### **3 THE C-K THEORY OF DESIGN**

C-K theory (Hatchuel and Weil, 2009) is a general descriptive model with a strong logical foundation, resulting in powerful expressive capabilities. The theory models design as interplay between two spaces, the space of concepts (C-space) and the space of knowledge (K-space). Four operators allow moving between and within these spaces to facilitate a design process:  $C \rightarrow K$ ,  $K \rightarrow C$ ,  $C \rightarrow C$  and  $K \rightarrow K$ . Space K contains all established, or true, propositions, which is all the knowledge available to the designer at any given moment. Space C contains "concepts", which are undecidable propositions (neither true nor false) relative to K, that is, partially unknown objects whose existence is not guaranteed in K. A concept is a hypothesis of the following form: "*there exists an entity  $x$ , for which the attributes  $A_1, A_2, \dots, A_i$  are true in K*".

Design processes aim to transform undecidable propositions into true ones by jointly expanding spaces C and K through the action of the four operators. This expansion continues until a concept becomes an object that is well defined by a true proposition in K. Expansion of C yields a tree structure, while that of K produces a more richly-networked pattern. This short introduction already shows that C-K theory provides a representation of the imaginable "states" in its C-space, and this representation happens to have a tree-shape, just like the structure of the state space in B&B. Moreover, C-K theory tracks in K-space the knowledge expansion; i.e., all the knowledge acquired and used during the design process. In particular, the evaluation criteria of the product to be designed are stored and enriched in K-space. Hence C-K theory appears to be a powerful framework to interpret design activities.

### **4 INTERPRETATION OF PARAMETER ANALYSIS WITH THE C-K FRAMEWORK**

A detailed account of interpreting PA by C-K theory is presented in (Kroll et al., 2014) and will be described here only briefly. It consists of translating each of PA's steps—PI, CS and E—to a combination of C-K operators. This interpretation is based on the fact that because knowledge is not represented explicitly in PA and because a design should be considered tentative (undecidable in C-K terms) until it is complete, both PA's parameters and configurations are entities of C-K's C-space. Thus, the description of a C-K concept has been refined from having just a list of attributes to having two kinds of attributes: ideational (corresponding to PA's parameters) and structural (PA's configuration items). The general form of a concept in C-K becomes: "*there exists an entity  $x$ , for which the ideational attributes  $P_1, P_2, \dots$  can be realized with the structural attributes  $S_1, S_2, \dots$* ".

The creative synthesis (CS) step starts with an idea, a PA parameter, and results in a new configuration through a "visit" to K-space (Figure 1a). A basic premise of PA is that ideas cannot be directly evaluated in an effective manner; rather, they need to be implemented as configurations first. So, the evaluation (E) step (Figure 1b) begins with a C-K concept that includes structural attributes and attempts to deduce its specific behavior, from which it will make a decision as to how to proceed. Reasoning from behavior to decision, however, includes two intermediary steps that are the key to understanding how the evaluation controls the design process so that it always moves in the most promising direction. First, the specific behavior is used to establish possible new evaluation criteria, and those are applied (together with the existing, older ones) to all pending concepts to assign values to them. The values are not numerical, as B&B's costs, but rather a metric that represents the designer's judgment of the goodness and viability of the concept. Finally, a decision is made to move so as to maximize this value.

The identification of new evaluation criteria is part of the actual learning done during the design process and is facilitated by having configurations to be evaluated. The combination of CS and E steps allows discovering unexpected behavioral aspects, or revealing that some known functional issues

have become more critical. New and critical issues in PA form the basis for the next PI step, as explained below, depending on the following four possible evaluation results:

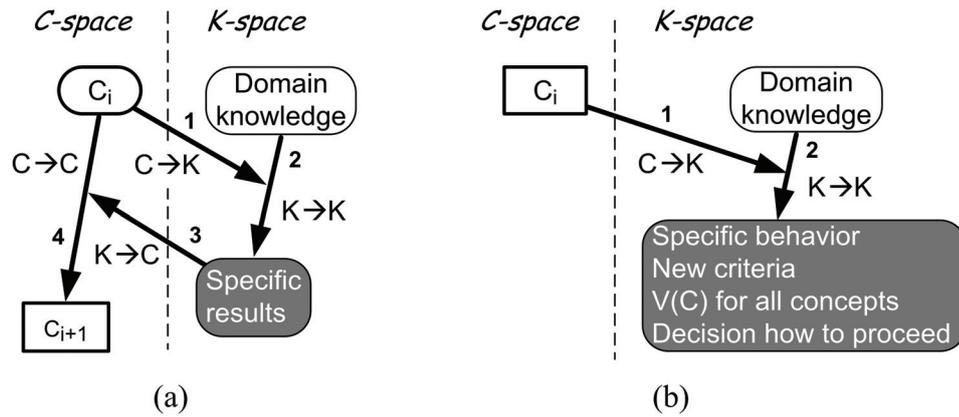


Figure 1. Interpreting (a) creative synthesis (CS) and (b) evaluation (E) as sequences of C–K operators. Arrow numbers denote the order; rounded-corner boxes denote concepts generated by adding an ideational attribute (PA parameter) and square boxes stand for concepts with an added structural attribute (PA configuration)

1. If the concept’s behavior is as desired and nothing is missing, and the value of the concept is higher than that of any other concept, then the design process is complete. All current attributes of the concept are accepted, and there is no subsequent PI step.
2. If an undesired behavior is detected, or something is missing in the concept, but its value is still the highest, then it should be improved by keeping its current attributes and adding a new ideational attribute in the next PI step (this is the most common occurrence), thus staying on the current path, as shown in Figure 2a.
3. If the undesired behavior renders another existing concept more valuable (perhaps because the evaluation criteria have changed), then the current development path should be stopped and backtracking to a known but unexplored path will take place, so the next PI step will continue with the new highest-value concept (Figure 2b).
4. If the value of all existing concepts and technologies is very low, then all their attributes should be rejected and backtracking to an unknown path will occur. The subsequent PI step will attempt to discover a new path by looking for a new core technology that had not been considered earlier (Figure 2c) or by re-examining the validity of the root concept itself, leading to a possible de-partitioning—the emergence of a more general start node (Figure 2d), as demonstrated in (Kroll et al., 2014).

## 5 DEMONSTRATION WITH A CASE STUDY

The following real design task had originated in the defense industry. The design process was reconstructed as PA by one of the inventors, the second author of this paper (Weisbord and Rotkopf, 2013). The need was to develop an airborne warhead for destroying concrete structures and it was required to be integrated into a given flight-certified aerodynamic configuration called “airborne body”, with fixed aerodynamic and other characteristics.

A need analysis stage defined the task as penetration and detonation of the explosive charge inside the structure. For a given concrete strength, the following engineering requirements were generated (bold letters stand for actual numerical values):

1. Required concrete penetration depth – **t** [m].
2. Minimum required weight of explosive – **W<sub>e</sub>** [kg].
3. Given the airborne body impact angle relative to the horizon, **β**, the design should operate properly within an angle-of-attack range of **±α** [°] relative to **β**.
4. The maximum terminal velocity of the airborne body is **V<sub>t</sub>** [m/s] and it is assumed to also be the impact velocity.
5. Maximum weight is **W<sub>b</sub>** [kg] (derived from the aircraft’s maximum takeoff weight).
6. Airborne body length is **L<sub>b</sub>** [m] and external diameter is **D<sub>b</sub>** [m], as in Figure 3a.

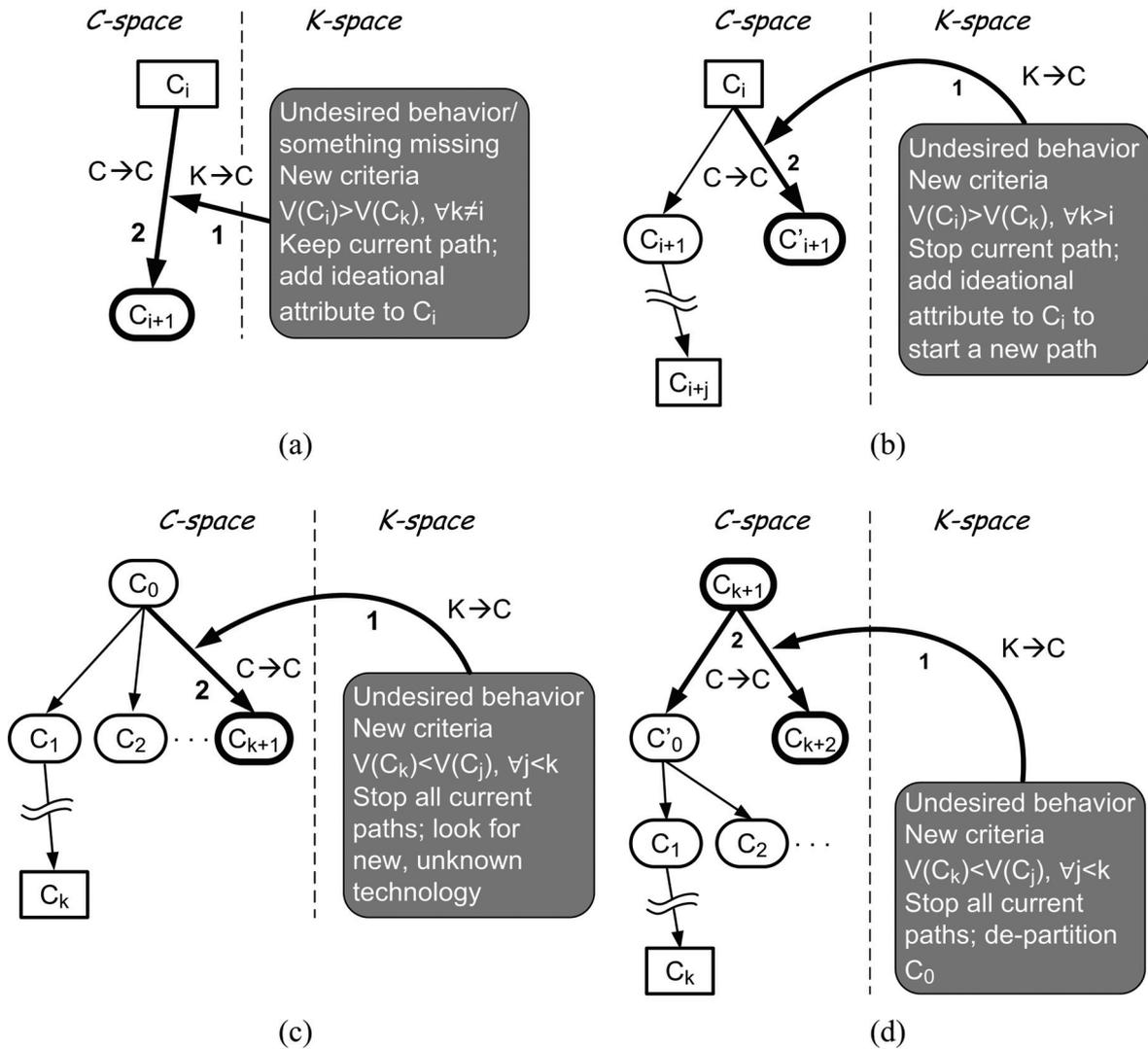


Figure 2. Interpreting PI as a sequence of C–K operators: (a) moving along the current path; (b) moving to a known but unexplored path; (c) starting over with a new, unknown technology; (d) de-partitioning the original root concept

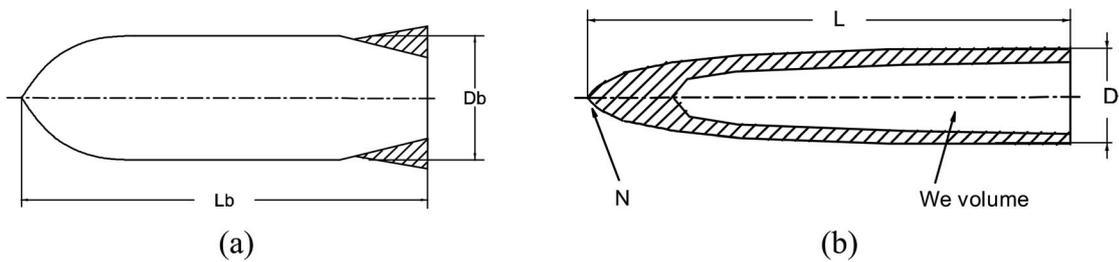


Figure 3. (a) Airborne body geometry; (b) a typical penetration projectile configuration ( $L$ -length,  $D$ -outside diameter,  $N$ -nose performance coefficient,  $We$ -explosive weight)

In what follows we use C–K theory’s terminology to describe the unfolding of the design process, so the word ‘concept’ will be used for all the designed artifacts, whether they have been created by adding ideational or structural attributes (see also the caption of Figure 1).

Technology identification (TI) began with the designers identifying the most difficult aspect of the design task as  $C_0 = penetrating\ concrete\ structures$  (and not, for example, detonation of the explosive charge, designing a safe and arm mechanism, or the method of deployment). Physical principles and core technologies for realizing the task were identified next. One of the well-known principles is based

on converting kinetic energy to penetration work as formulated by Young's model (Young, 1997), in which the penetration depth  $t$  is given as (shown simplified)  $t \propto N \cdot (W/A) \cdot V$ , where  $N$  is a dimensionless nose performance coefficient,  $W/A$  is the projectile's weight-to-frontal area ratio, and  $V$  is the impact velocity. Choosing this principle for the design task at hand meant that the solution concept became  $C_1 = \text{converting kinetic energy to work by a penetration projectile}$ . In a typical penetration projectile (Figure 3b), the internal cavity contains the explosive, so at the end of the penetration process, and provided that the projectile did not break (one of the challenges), the explosive detonates.

Being familiar with penetration projectiles, the designers realized that there were two possible concepts they could use:  $C_2 = \text{an airborne body shaped as a penetration projectile}$ , and  $C_3 = \text{a penetration projectile packaged inside an airborne body}$ . Reviewing some pros and cons,  $C_2$  was chosen for further development. This terminated the TI stage and constituted the initial conditions for PA.

This first parameter, idea, was realized by setting the length, diameter and explosive weight according to requirements 6 and 2, respectively ( $C_4$ ), as shown in Figure 4a. Evaluation next showed that the projectile would not penetrate the required concrete thickness, and that  $N$ ,  $W/A$  and  $V$  could not be increased because they were fixed by the given airborne body. The decision made was to backtrack to  $C_3$  from the TI stage so as to increase  $W/A$  and  $N$ , and this yielded concept (in fact, what is a 'configuration' in PA)  $C_5$ , where a smaller  $D$  and heavier metals increased  $W/A$  and a pointed nose increased  $N$  (Figure 4b). Evaluating  $C_5$ , it was found that penetration performance was better, but still did not meet the requirement. Analysis showed, however, that increasing the impact velocity by a certain  $\Delta V$  could solve the problem. So, the next parameter (ideational attribute) was "increase the impact velocity by a rocket engine" ( $C_6$ ). The corresponding realization ( $C_7$ ) added structural attributes in the form of a specific rocket engine fitted to the back of the projectile and some dimensions that were adjusted accordingly, as shown in Figure 4c.

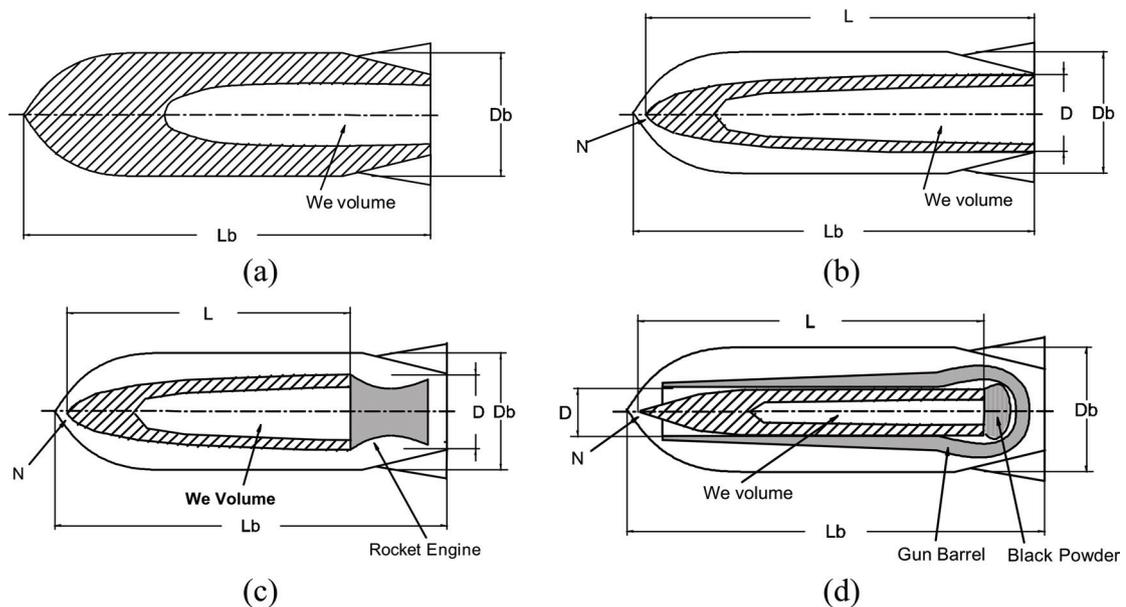


Figure 4. Concepts  $C_4$  (a),  $C_5$  (b),  $C_7$  (c) and  $C_9$  (d). All are 'configurations' in PA terms

The subsequent evaluation found that penetration performance was satisfactory, but two new problems emerged: rocket ignition had to be done at high altitude to allow the relatively long burn-time, potentially causing deviation of the trajectory from the required impact angle; and the amount of explosive was now less than the required  $We$ . Although adding stabilizers and/or a guidance unit might solve the problems, it would increase the cost, complicate the product and reduce its reliability. Another way to increase the impact velocity was needed. The designers responded with the idea of accelerating the projectile with a gun barrel ( $C_8$ ), which was realized as the configuration of Figure 4d ( $C_9$ ). Calculations and simulations showed that the penetration performance was as required, and burn-time of the gunpowder was short so ignition could take place at low altitude (short distance to the target). However, the amount of explosive was now lower due to the reduced projectile diameter. The

designers discussed the options with the customer and decided to reduce the explosive amount and carry on with further development of this solution. Major design challenges that remained were a strong, yet lightweight gun barrel, an accurate means of measuring altitude above the target to fire the projectile, and an interface between the gun barrel and the airborne body to prevent sideways movement during recoil.

The re-negotiation of requirement 2 that took place demonstrated that in a real design task, unexpected requirements and constraints may appear *after* the designers generate and evaluate solution alternatives, thereby increasing their (and perhaps also the customer's) understanding of the problem and the scope of the design task. This well-known phenomenon is in-line with the often ill-defined nature of design problems and the necessity to evolve their definition together with the solution (Dorst and Cross, 2001; Maher and Tang, 2003). It also supports our earlier claim that design cannot be treated as mere search.

Figure 5 shows the concept tree generated in C–K theory's C-space for the case study. The root concept  $C_0$  led to several possible technologies:  $C_1$  was chosen for further development and the other, unexplored technologies are represented as a dashed arrow. The concept tree follows the above description, ending in concept  $C_9$ .

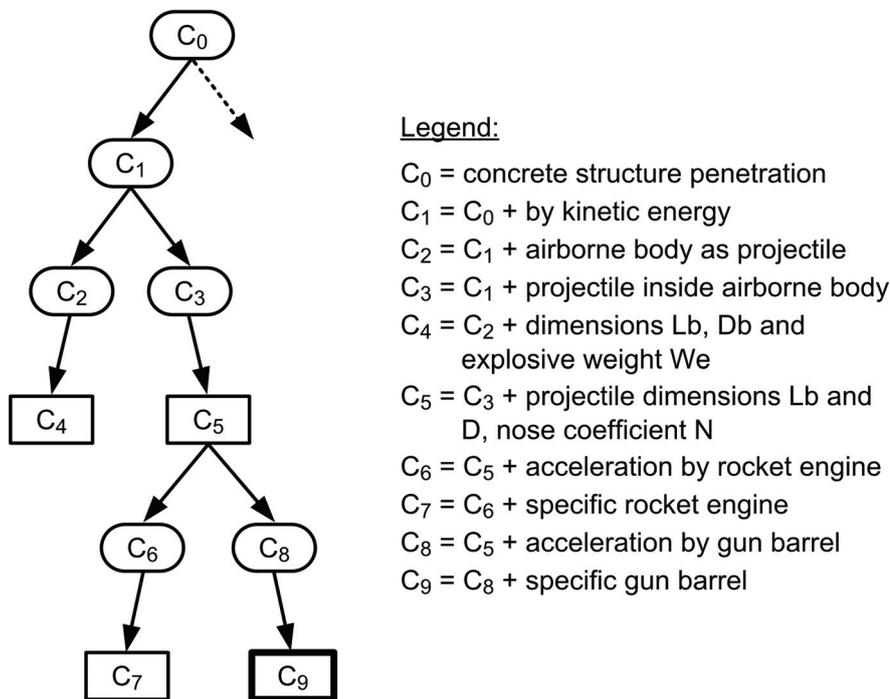


Figure 5. The concept tree generated for the case study.  $C_9$  was the solution generated

## 6 DISCUSSION AND CONCLUSION

PA appears to be an odd combination of design and search. It is a design process in the sense that there is no target solution at the beginning and surprises and discoveries are expected at each step, particularly through the evaluation of configurations. But its reasoning and strategy also share many features with B&B methods: PA incorporates opportunities and activities of diverging that seem similar to B&B's branching, and PA relies heavily on constantly evaluating the artifact, and this is analogous to B&B's bounding by a cost function. Hence, studying PA helps to understand how B&B can be extended to design processes. To make this extension rigorous, we use a design theory, C–K, to better follow how PA actually helps to navigate strategically in the unknown state space and unknown goal state, just as B&B helps to traverse the space of complex optimization problems (with complex but known state space and goal state).

Although we used a specific conceptual design method, its activities are generic: coming up with solution ideas, implementing them as configurations, evaluating the evolving design, comparing alternatives, and choosing the most promising one. Therefore, it seems plausible that other design methods can be represented as generalized B&B search as well.

The extensions to the “classical” B&B method that are required in order to generalize it so it can be applied to design processes are therefore as follows:

1. Allow flexibility in defining the start node,
2. Allow unknown goal states,
3. Allow “branching” by design moves that are not pre-defined,
4. Allow a non-numerical evaluation function (the “bounding”) that is subject to learning during the design process,
5. Allow sub-tree “pruning” to be temporal (i.e., a pruned branch may be revisited later, as the design process unfold, if new knowledge is acquired).

Recall the model of innovative design proposed by Kazakçi (2014), which allows the creation of new types of objects and new ways to construct them. This duality seems to resemble PA’s two synthetic moves: PI for adding ideational attributes to the artifact (i.e., changing the type of object), and CS for adding structural attributes (that is, coming up with the methods to realize the new type). Moreover, just as we have proposed to extend the conventional search and optimization methods to make them usable in design, Kazakçi too calls for a new framework (“Brouwer machine”) that allows “free choices”, and whose purpose is not to choose a best alternative among existing objects, but rather to choose the most interesting to explore among a set of novel definitions.

There are several possible contributions and advantages that could result from considering conceptual design as extended and generalized B&B search. First, design processes need a strategy (that is, the designer needs to know how to proceed at any moment), and both AI and OR provide many well-established and much-studied strategies. In the case of PA, we call it “steepest-first”; it is a depth-first with backtracking type of process, but the PI step controls it by addressing the more difficult and challenging issues first. The generalized B&B modeling thus helps to understand how PA provides guidance in the concept tree of C-space towards the goal.

Secondly, computer programs use a clear logic and data representation according to which they operate, and these may be useful in explaining human design processes. Instead of analyzing design methods directly, we use an intermediate representation of the artifact based on the tree structure of C–K’s C-space. This allows a convenient analogy to similar structures in B&B and AI-based searches. It would be interesting in the future to apply similar modeling to other design methods and compare them. A third contribution of looking at design from the AI perspective may be in building computerized design assistants. These programs may not need to be totally autonomous, of course, but rather guide the human designer in following the desirable strategy and managing and organizing some of the complexities of the large search spaces. Finally, modeling the design process as computer-based extended B&B can serve as the basis for a tool for capturing design rationale.

Some possible limitations of our work should also be noted. The example of this paper and other examples we studied tend to be quite simple, for convenience of presentation. Real-life situations are of course much more complex, and we cannot predict the consequences of applying the same modeling framework to cases where many goals may contradict and interact with each other. We also have not explored issues related to collaborative design, which may lead to further complications. Another weakness may also be the need to evaluate the evolving design artifact quantitatively. This “cost” or “value” function is quite straightforward notionally, but may be difficult to implement in practice.

## REFERENCES

- Antonsson, E.K. and Cagan, J. (eds) (2001) *Formal Engineering Design Synthesis*. Cambridge: Cambridge University Press.
- Badke-Schaub, P. and Eris, O. (2014) A theoretical approach to intuition in design: Does design methodology need to account for unconscious processes? In Chakrabarti, A. and Blessing, L.T.M. (eds), *An anthology of theories and models of design: Philosophy, approaches and empirical explorations*, London: Springer-Verlag, Chapter 17.
- Cagan, J., Grossmann, I.E. and Hooker, J. (1997) A conceptual framework for combining artificial intelligence and optimization in engineering design. *Research in Engineering Design*, Vol. 9, No. 1, pp. 20-34.
- Dorst, K. and Cross, N. (2001) Creativity in the design process: Co-evolution of problem–solution. *Design Studies*, Vol. 22, No. 5, pp. 425-437.
- Dym, C.L. and Brown, D.C. (2012) *Engineering Design: Representation and Reasoning*. 2nd ed., Cambridge: Cambridge University Press.

- Gries, M. (2004) Methods for evaluating and covering the design space during early design development. *INTEGRATION, the VLSI journal*, Vol. 38, No. 2, pp. 131-183.
- Hatchuel, A. (2001) Towards design theory and expandable rationality: The unfinished program of Herbert Simon. *Journal of Management and Governance*, Vol. 5, No. 3-4, pp. 260-273.
- Hatchuel, A. and Weil, B. (2002) C–K theory: Notions and applications of a unified design theory, Herbert Simon International Conference on Design Sciences, Lyon, March 15-16.
- Hatchuel, A. and Weil, B. (2009) C–K design theory: An advanced formulation. *Research in Engineering Design*, Vol. 19, No. 4, pp. 181-192.
- Hillier, F.S. and Lieberman, G.J. (2005) *Introduction to Operations Research*. 8th ed., New York: McGraw-Hill.
- Kazakçi, A.O. (2014) Conceptive artificial intelligence: Insights from design theory, International Design Conference - DESIGN 2014, Dubrovnik, May 19-24, University of Zagreb/The Design Society.
- Korf, R.E. (2010) Artificial intelligence search algorithms. In: Atallah, M.J. and Blanton, M. (eds), *Algorithms and theory of computation handbook: Special topics and techniques*, 2nd ed., Boca Raton: Taylor and Francis, Chapter 22.
- Kroll, E. (2013) Design theory and conceptual design: Contrasting functional decomposition and morphology with parameter analysis. *Research in Engineering Design*, Vol. 24, No. 2, pp. 165-183.
- Kroll, E., Condoor S.S. and Jansson, D.G. (2001) *Innovative Conceptual Design: Theory and Application of Parameter Analysis*. Cambridge: Cambridge University Press.
- Kroll, E., Le Masson, P., and Weil, B. (2014) Steepest-first exploration with learning-based path evaluation: Uncovering the design strategy of parameter analysis with C–K theory. *Research in Engineering Design*, Vol. 25, No. 4, pp. 351-373.
- Li, X., Zhang, Z-N., Liu, Z-L. and Xie, Y-B. (2013) A novel semi-heuristic planning approach for automated conceptual design synthesis. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 227, No. 10, pp. 2291-2305.
- Maher, M.L. and Tang, H.H. (2003) Co-evolution as a computational and cognitive model of design. *Research in Engineering Design*, Vol. 14, No. 1, pp. 47-63.
- Mitchell, W.J. (2001) Vitruvius redux: Formalized design synthesis in architecture. In: Antonsson, E.K. and Cagan, J. (eds), *Formal engineering design synthesis*, Cambridge: Cambridge University Press, Chapter 1.
- Nau, D.S., Kumar, V. and Kanak, L. (1984) General branch and bound, and its relation to A\* and AO\*. *Artificial Intelligence*, Vol. 23, No. 1, pp. 29-58.
- Pearl, J. (1984) *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Reading: Addison-Wesley.
- Russell, S.J. and Norvig, P. (1995) *Artificial Intelligence: A Modern Approach*. Upper Saddle River: Prentice Hall
- Simon, H.A. (1969) *The Sciences of the Artificial*. Cambridge: MIT Press.
- Simon, H.A. (1972) Theories of bounded rationality. In: McGuire, C.B. and Radner, R. (eds), *Decision and organization*, Amsterdam: North-Holland, pp. 161-176.
- Tong, C. and Sriram, D. (1992) *Artificial Intelligence in Engineering Design (Vol. I): Design Representation and Models of Routine Design*. San Diego: Academic Press Professional.
- Weisbord, G. and Rotkopf, M. (2013) Shell accelerator. International patent application no. PCT/IB2013/054657.
- Young, C.W. (1997) Penetration equations. Sandia National Laboratories report no. SAND97-2426.

## ACKNOWLEDGMENTS

This work was supported by the Israel Science Foundation under Grant No. 546/12.