Proceedings of the Design Society, Volume 5: ICED25
https://doi.org/10.1017/pds.2025.10260

Modeling constraint-based manufacturing networks: theory, application and best practices

Kevin Herrmann ✉, Felix Pusch, Christian Becker, Paul Christoph Gembarski and Roland Lachmayer

Leibniz University Hanover, An der Universität 1, 30823 Garbsen, Germany

✉ herrmann@ipeg.uni-hannover.de

ABSTRACT: This paper demonstrates how the Portfolio of Capability Constraint Network (PCCN) facilitates modeling and analyzing complex manufacturing networks by framing them as constraint satisfaction problems (CSPs). These models face high complexity due to numerous n-ary constraints and large solution spaces, posing challenges for standard solution algorithms. Existing CSP remodeling approaches were reviewed but found unsuitable for the specific needs of PCCNs. As a result, tailored design guidelines and heuristics were developed to reduce problem complexity effectively. The applicability of these guidelines was validated using a use case involving the production of a multi-material shaft with tailored forming technology. Results showed significant efficiency gains in solution searches, emphasizing the practical value of the proposed methods in simplifying and optimizing PCCN-based models.

KEYWORDS: Design for X (DfX), Constraint modeling, Knowledge management, Tailored forming, Remodeling

1. Introduction and motivation

In operations research and engineering, mathematically modeling specific problems for computer-aided analysis and solution finding is a major challenge (Domschke et al., 2015). This requires both a deep understanding of the problem and knowledge of mathematical methods for efficient modeling. Practical feasibility with computer tools and the speed of finding solutions are key concerns (Kumar, 1992; Russell and Norvig, 2010).

In production engineering, determining whether a specific component can be manufactured is complex (Pahl et al., 2007; Wawer et al., 2023; Groche et al., 2017). This requires formalizing manufacturing capabilities and linking them to component geometric models (Gembarski, 2020). Components are produced through coordinated processes in a manufacturing chain, making holistic modeling essential, as analyzing individual processes in isolation is insufficient (Herrmann et al., 2021, 2023a). Many studies on manufacturability focus on near-net-shape components, overlooking emergent constraints (Albrecht and Anderl, 2016; Nellippallil et al., 2018; Anjum et al., 2012). This research gap is addressed by Herrmann et al. (Herrmann et al., 2023a) through the Portfolio of Capability Constraint Network (PCCN), which frames manufacturability analysis as a configuration problem of manufacturing stages. This configuration problem is formulated as a Constraint Satisfaction Problem (CSP) (Herrmann et al., 2023b). However, the PCCN includes many variables with big domains which lead to an extensive solution space that needs to be searched. In addition, it contains a large number of constraints, many of which are com-pex n-ary constraints that represent the real relationships within the manufacturing process. This means that the PCCN presents a challenge for existing algorithms like constraint propagation, backtracking, and forward-checking, and leads to long program runtimes. There are two ways to overcome these challenges: (1) developing a specialized solution algorithm using heuristics and a problem-specific approach, and (2) reducing problem complexity through a more solver-friendly formulation for use with existing algorithms (Bessière, 1999).

This article presents design guidelines and heuristics for modelling a PCCN in order to reduce problem complexity and create an optimised representation for solution algorithms. Modellers are methodically supported and the efficiency of problem solving is increased. Validation is based on a use case that investigates the manufacturability of a multi-material shaft using tailored moulding technology. The generalised problem formulation of the PCCN enables the findings to be transferred to other components and process chains. The PCCN complexity is analysed scientifically, while the practical contribution offers applicable guidelines and heuristics for reducing the complexity of the problem.

2. Research background

2.1. Constraint satisfaction problems

In constraint programming, problems are defined as Constraint Satisfaction Problems (CSP), where variables, domains and constraints are described declaratively (Dechter, 2003). Sequencing or configuration problems often require an abstraction of the original problem (Brailsford et al., 1999; Gemabarski, 2022). CSPs belong to the NP complexity class, which leads to large search spaces for practically relevant problems (Bulatov, 2017; Zhuk, 2020). Algorithms such as backtracking are supplemented by constraint propagation and heuristics to maintain local consistencies (Wegener, 2005; Russell and Norvig, 2010). Examples of this are forward checking and maintaining arc consistency (MAC) (Sabin and Ereuder, 1997; Jussien et al., 2000). Local search algorithms such as the Minimum Conflict Algorithm also play a role (Clark et al., 1996). However, global inconsistencies can only be recognised late, which makes the solution more difficult (Galinier and Hao, 2004). Well-known algorithms such as AC-3 are effective for binary CSPs (Mackworth, 1977; Ruttkay, 1998), while n-ary constraints with non-linear expressions, which frequently occur in practical problems, lead to high complexity (Bessière, 1999; Lhomme and Regin, 2005). Generalised Arc Consistency (GAC) and Hyperarc Consistency (HAC) extend the AC family for n-ary constraints, but remain computationally intensive (Regin, 1996; Bergenti and Monica, 2017). In addition to more efficient algorithms, there are remodelling approaches for reducing problem complexity (Apt, 2003; Marriott and Stuckey, 1998). These include: (1) Binary transformations that convert each CSP into a binary CSP with constant expressive power (Rossi et al., 1990), e.g. through dual or hidden transformations (Dechter and Pearl, 1989; Bacchus et al., 2002). (2) SAT transformations in which variables and constraints are converted into Boolean problems and powerful SAT solvers are used (Petke, 2015; Walsh, 2000). (3) Tabulation and regularisation, in which substitutable subproblems are identified and converted into efficiently solvable table or regular constraints (Akgün et al., 2018; Trick, 2003). These approaches offer targeted speed advantages, but require additional effort for the transformation.

2.2. Portfolio of capability constraint network

The PCCN is a constraint-based modeling approach that aims to connect the idealized solution space of a component with the real solution space of manufacturing. This model enables the analysis and optimization of the manufacturability of a component design along an entire manufacturing process chain. By explicitly modeling a manufacturing process chain, the PCCN allows for the consideration of emergent manufacturing constraints and thus differs from related work.

To model a PCCN, information about the manufacturing process chain that will be used to manufacture the component is required in addition to the component design considered (Herrmann et al., 2023a). For this, the process parameter windows that lead to the production of high-quality products must be sufficiently known. In addition, the manufacturing process chain must be mastered, i.e., the process behavior must be predictable (Sheveleva et al., 2023). Modeling a PCCN can be divided into five phases:

1. **Analysis of the Process Chain:** The first phase consists of analyzing the manufacturing process chain to identify all subprocesses, process parameters, production stages and available resources such as machines or tools.
2. **Grouping of Process Containers:** In the second phase, all recorded entities in the form of subprocesses, process parameters, production stages, and resources are grouped into process containers. The PCCN divides a manufacturing process chain into independent process containers that describe the transformation of manufacturing steps. A process container can be thought of as a black box into which a production step enters and a transformed production step exits.

3. **Parameterization:** In the third phase, parameters are created to abstract the incoming and outgoing production stages, the resources involved, and the necessary process parameters for constraint-based modeling based on the process containers. These parameters are the variables of the PCCN and are then completed with a domain assignment.
4. **Modeling the Transformative and Restrictive Constraints:** In the fourth phase, transformative and restrictive constraints are formulated for the process containers. The transformative constraints link all the parameters of the incoming and outgoing production stages of the process container, and thus describe the transformation of a production stage within a process container. Restrictive constraints limit the possible values of the production stage parameters, and thus describe the production capabilities of the equipment used.
5. **Merging the Constraint Network:** In the final stage, the transformative and restrictive constraints of all process containers are combined to form a constraint network.

The modeling of a process container is demonstrated using the example of a friction welding process (see Figure 1). Friction welding is a welding process in which two cylindrical components are moved relative to each other under pressure. The resulting friction causes the contact surfaces to heat and the material to plasticize. The geometric shape of the two cylindrical input components can be mathematically described by a diameter d_1 and d_2, as well as a length l_1 and l_2. The same applies to the initial components joined with d_3 and l_3. The geometry of the weld bead, which is usually created, is neglected here because it is usually turned off immediately after friction welding. In addition, the length loss Δl_{bead} due to bead formation and a bead formation ratio φ_{bead}, which divides the length loss over the lengths of the input components $\varphi_{bead} = \frac{\Delta l_{1,bead}}{\Delta l_{2,bead}}$, are added as process variables. In friction welding, the two diameters of the input components must be equal ($d_1 = d_2$). Since the diameters of the components do not change during friction welding, they can be modeled as constant throughout the process ($d_1 = d_3$ and $d_2 = d_3$). To calculate the length l_3 of the joined component, the lengths of the input components l_1 and l_2 can be added and the length loss can be proportionally subtracted Δl_{bead}. If the bead formation ratio φ_{bead} is 1, the weld bead is formed only from the first component. At a ratio of 0, it is formed only from the second component, and at a ratio of 0.5, both components contribute equally to the formation of beads ($l_3 = (l_1 - (\phi_{bead} * \Delta l_{bead}) + (l_2 - (\phi_{bead} * (1 - \Delta l_{bead})))$). Finally, a process window is formulated for the diameters and lengths using minimum and maximum values that can be realized by a specific friction welding machine. Within the process window, a specific travel Δl_{move} of the friction welding machine is also required ($l_1 + l_2 + \Delta l_{move} > l_{min}$, $l_1 + l_2 + \Delta l_{move} < l_{max}$, $d_3 > d_{min}$ and $d_3 < d_{max}$).
The data structure of this process container can be seen as part of (Herrmann et al., 2024). Each variable has a name, a domain, that is assigned to it and an optional comment. The domain consists of one, multiple or a range of numbers. Each constraint has a number as a specific ID, a list of the involved

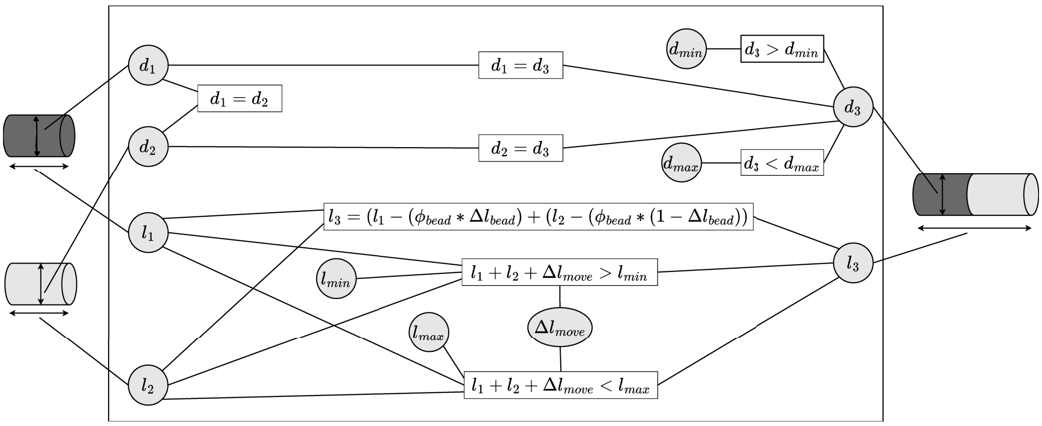

Figure 1. Process container friction welding

variables and the constraint-equation which are the important information for the CSP. In addition, each constraint has a source indicating whether it is a manufacturing restriction or a transformation. This information can be used for optional conflict solving, which is not discussed further here. Each constraint also has a comment and a process to which the constraint belongs. This is information that helps the user to understand what the constraint represents.

After modeling, general solution algorithms for CSPs can be used to search for a consistent configuration of all production steps. The PCCN is primarily not a replacement for production planning, such as creating work schedules or setup plans. The purpose of the PCCN is to help designers identify and resolve production conflicts early. The goal is to shorten development cycles between design and production planning, increase the efficiency of the product development process, and provide a better basis for production planning.

3. Problem analysis and challanges

The general structure of a PCCN is methodically supported by modeling techniques. However, this approach does not necessarily lead to a solver-compatible representation of the PCCN. The complexity of the CSP of a PCCN depends strongly on the modeled manufacturing process chain. For practical applications, this complexity exceeds the complexity of many academic benchmark problems, such as the Send-More-Money problem (Nareyek, 2001) or the Sudoku problem (Simonis, 2005). A PCCN combines several challenges for the solution process.

- As the complexity of the components and manufacturing steps considered in the PCCN increases, so does the number of parameters required to describe the geometries fully. Depending on the application, this leads to many variables in the CSP. In addition, a high degree of flexibility in the domains is required to cover a manufacturing process's high degrees of freedom. For example, considering the cutting of profile bodies, a 100 *mm* can be cut to any length between 1 *mm* and 99 *mm*, depending on the increment. The domain must contain the entire range as an interval. A large number of variables paired with large interval domains leads to a large search space, which must be checked when searching for a solution.
- In practice, manufacturing processes often involve complicated transformations of components that are nonlinear and difficult to model. For example, in impact extrusion, the formability of a part depends on several factors, including the blank geometry, the forming path, and the shape of the extrusion die. Mathematically describing these processes requires complicated nonlinear constraints that represent the behavior of the part under different conditions. In the case of impact extrusion, constraints on volume constancy, degree of deformation, and yield stress can be combined to model the process (Lange et al., 2008).
- In addition, the description of the complicated component transformations leads to constraints with a large number of bounded variables. This results in a non-bijective mapping of the transformations and a highly constrained and meshed CSP. As a result, the solution of one constraint often depends on the solutions of other constraints. This increases the complexity of the solution search since changes in one point of the mesh can affect other parts.
- This interdependence of constraint solutions is reinforced by the procedural structure of the PCCN. It has a globally directed structure that refects the sequential flow of production processes, but the local interactions between variables are represented by undirected edges. This means that the constraints of one process cannot be solved independently of the constraints of the other processes.

Against the background of the inherent complexity and specific problems of a PCCN, a differentiated and critical consideration of the applicability of remodeling approaches proves to be essential for an efficient search for solutions.

When transforming a PCCN into a binary CSP, the transformed domains grow exponentially with the large solution space and n-ary constraints, leading to high time complexity and memory consumption since the large solution space is explicitly modeled in the binary transformation (Tsang, 1995). Converting a PCCN into an SAT problem results in a large, complex SAT instance due to non-linear constraints, which require additional variables and statements, complicating solvability. Both binary and SAT solvers typically require transforming the entire problem, making the transformation effort often greater than solving the original CSP.

Tabularization and regularization can convert parts of a CSP into a more efficient representation. Since a PCCN is a highly interconnected CSP, identifying substitutable partial CSPs is challenging. The process-related representation of a manufacturing process chain means sub-CSPs cannot be considered independently, as they depend on each other sequentially. Therefore, solving them directly is impossible without redesigning the entire CSP. In a PCCN with many variables and large domains, tabulating partial CSPs results in enormous tables, which are impractical to create and store for many manufacturing process chains. Introducing regular constraints through regularization is an efficient way to reduce the complexity of constraints. A PCCN often contains various non-linear expressions in the constraints due to the complex mapping of component transformations. Describing these complicated relationships between variables with regular constraints is extremely difficult, if not impossible, without simplifying or omitting key aspects of the problem.

As shown by (Wallace, 2005), heuristics can contribute to a faster solution of a CSP. However, at a certain complexity of the CSP such as the PCCN these conventional heuristics which can be used for any CSP do not contribute enough to reduce the duration of the solution search sufficient enough.

In summary, the challenges of a PCCN make the mentioned transformation methods impractical or inefficient. Moreover, transformations reduce human readability and clarity of process relationships and must be repeated whenever the PCCN is extended or adapted, leading to high modeling costs. While it is theoretically possible to model the PCCN as a special form of a CSP (regular, table, binary, or Boolean), this complicates the modeling process to the point of impracticality. Typically, the goal is to model a constraint problem as freely and closely to the application as possible (Löffler, 2022). Since transforming a PCCN into another formulation does not lead to a more efficient solution search, complexity reduction can only occur at the same representation level. This requires examining the explicit formulation of variables and constraints in a solver-friendly way and developing problem-specific heuristics for a more efficient solution search.

4. Design guidelines and heuristics

Due to the large solution space of a PCCN, the use of constraint propagation algorithms is necessary, since otherwise, a search algorithm must examine, in the worst case, n variables with k elements each in the domain of k^n assignments of variables to search the search space for a consistent solution (Mackworth, 1977; Brailsford et al., 1999). In the case of the PCCN with its n-ary constraint, the constraint propagation algorithm must also be able to propagate hyperedges(Regin, 1996). The GAC algorithm can be used for this purpose. If r describes the maximum arity of a constraint, e the number of constraints, and d the maximum number of elements in a domain, the time complexity of the GAC algorithm is $O(e * r * d^r)$ (Lecoutre, 2008). The main challenge in this context is the increased computational complexity, which is influenced by e, r and d. These parameters are determined by the mathematical formulation of the problem and can be reduced by clever problem formulation. Therefore, the following design guidelines support the reduction of computational complexity. The specific challenges of a PCCN are addressed. The table 1 summarizes all design guidelines with examples.

1. When modeling component transformations across multiple manufacturing processes, the parameters of one component may be not changed. In this case, the parameters are passed from one manufacturing step to the next using the "=" operator. This type of modeling causes the number of constraints to increase. This situation can be resolved by combining parameters that remain the same across multiple production stages into a common variable (see table 1 (1)). This reduces the number of constraints (e) that need to be checked and analysed, thus speeding up the solution search process.
2. When manufacturing constraints are integrated directly into the PCCN by numerical value, unary constraints are created. An example is the integration of a critical strain $\phi > 0.7$. Such unary constraints can be considered directly in the domains of the variable involved (see table 1 (2)). This also reduces the number of constraints (e). At the same time, however, the domains (d) are also restricted in advance, which reduces the search space.
3. Especially with complicated component transformations, many variables influence a constraint, which leads to high articity. This articity is exponentially included in the time complexity of the GAC algorithm. Therefore, in many cases, it makes sense to divide long constraints with many variables into several smaller constraints using auxiliary variables. Although this increases the number of variables and constraints, the exponential influence of articity (r) can be reduced (see table 1 (3)). This reduces the recursive loops in the solution search.
4. When mapping the transformations from one production stage to the next, the transformation must be fully defined. This means that all parameters of the involved production stages must be included in at least one constraint. If a parameter is not constrained, it is irrelevant to the transformation and can be removed (see table 1 (4)).
5. Due to the partially non-linear expressions within constraints, calculations can lead to real results, e.g. $log(4) \approx 0.60206$. However, a PCCN, as a finite CSP, is not able to map real numbers due to the incremental division of the domains. This can mean, for example, that an equality between a

domain value and a real number can never be satisfied, $\forall a \in \{0.5, 0.6, 0.7\} \rightarrow a \neq log(4)$. For many manufacturing processes, the transformations can also be approximated. This means that equality relations with real numbers can also be simplified by rounding ($a \approx log(4)$) or a tolerance range ($a > log(4) * 0.9 \wedge a < log(4) * 1.1$) (see table 1 (5)).
6. The interval-like mapping of domains results in a large search space of a PCCN. Therefore, when modeling domains, care must be taken to keep the intervals as narrow and application-oriented as possible (see table 1 (6)). This reduce the maximum number of domain entries (d).

In addition to the design guidelines, two PCCN-specific heuristics can be used to make the solution process more efficient.

1. Due to the procedural nature of the PCCN, the assignments of variables build on each other. This means that a PCCN can be solved particularly efficiently if the individual production stages are configured in the inverse direction, i.e., starting with the last production stage n, through production stage $n-1$ to the first production step 1. This can be implemented by sorting the variables in descending order by their production stage affiliation. This avoids inconsistent configurations of the upstream production stages, which can lead to unresolvable conflicts that are only detected at a later stage.
2. Sorting is also useful for the order of constraints. First and foremost, constraints should be sorted according to their process container affiliation, such as variables in descending production direction. Within the process containers, the restrictive constraints should be checked for consistency before the transformative constraints, since a transformation is only possible if the production capabilities allow it.

Both heuristics can either be considered manually during modeling or implemented automatically using known sorting algorithms. The application of the design guidelines and the PCCN-specific heuristics leads to a significant increase in the efficiency of the solution process. This is demonstrated in the following use case.

5. Use case study

Tailored forming is an innovative manufacturing technology for producing solid components like shafts or gears from different materials. This technology and multi-material design enable highly efficient components (Ashby and Cebon, 1993; Herrmann et al., 2022). Tailored forming follows a multi-stage manufacturing process chain of joining, forming, and final processing. In this application example, the specific manufacturing process for a multi-material shaft is instantiated as a PCCN. The process begins by turning a cone and counter-cone in two mono-material cylinders, e.g., steel alloys (20MnCr5 or 100Cr6) and aluminum alloy EN AW 6082 (see figure 2). These conical shapes increase the joining surface and strengthen the joint. Next, the parts are joined by friction welding. The hybrid semi-finished product is then thermomechanically processed via extrusion. Finally, machining shapes the shaft and optimizes edge zones and surfaces. Additional functional elements, such as key or retaining ring grooves, are integrated at this stage.

Table 1. Design Guidelines.

Number	Design Guideline	Measure	Unfavorable	Better
(1)	Avoidance of constraints for the representation of constant variables	Summary of constant variables along the process chain into one variable	$a_1, a_2 : \{1,2,3\}$ $a_1 = a_2$	$a : \{1,2,3\}$
(2)	Avoidance of unary constraints	Integration of the constraint directly into the domain of the respective variable	$a : \{1,2,3\}$ $a < 3$	$a : \{1, 2\}$
(3)	Avoidance of constraints with high arity	Splitting of constraints and introduction of auxiliary variables	$a,b,c,d,e : \{1,2,3\}$ $a + b - c = d - e$	$a,b,c,d,e : \{1,2,3\}$ $h_1,h_2 : \{0,1,2,3,4,5\}$ $h_1 = a + b - c$ $h_2 = d - e$ $h_1 = h_2$
(4)	Avoiding open degrees of freedom in the transformations	Remove unconstrained variables	$a,b,c : \{1,2,3\}$ $a = b$	$a,b : \{1,2,3\}$ $a = b$
(5)	Avoid floating point numbers in the equation results	Include tolerance range or rounding in the constraints	$a : \{0.5,0.6,0.7\}$ $b : \{3,4,5\}$ $a = log(b)$	$a > log(b) * 0.9$ $a < log(b) * 1.1$ or $a \approx log(b)$
(6)	Avoiding domains that are too large	Dimensioning the domains close to the problem	$a : \{1, ..., 1000\}$	$a : \{100, ..., 150\}$

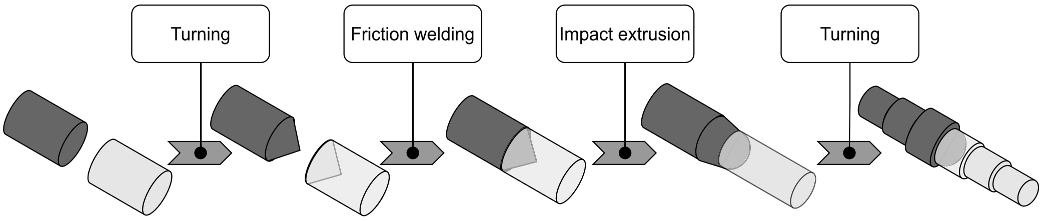

Figure 2. Process Chain for the manufacturing of a multi-matial shaft

By section 2.2, the manufacturing process chain was modeled as a PCCN using application-oriented constraints. The specific PCCN modeling was published in (Herrmann et al., 2024) and is not discussed further here. The design guidelines from the table 1 were then applied, and the PCCN was remodeled. Table 2 provides examples. Six variables were merged as they represent a constant diameter across multiple processes (guideline (1)). Additionally, a variable's domain was adjusted to a more application-relevant range (guideline (6)). Finally, guideline (5) was applied to determine the angled part's length after impact extrusion using a finite domain representation. The optimized PCCN of the manufacturing process chain was also published in (Herrmann et al., 2024) for transparency of the results.

The table 3 shows a comparison between the original and the optimized PCCN. The PCCN could be reduced by ten variables and constraints due to the more efficient representation method. This restricts the solution space by 19 powers of ten and reduces the mathematical complexity of the problem.

To quantify this efficiency gain, both PCCNs are solved using a constraint solver. The solver works in two stages and combines constraint propagation in preprocessing and a subsequent solution search using a forward-checking algorithm. Since the instantiated PCCN receives n-ary constraints, the GAC algorithm

Table 2. Remodeling of the PCCN

Guideline	Original PCCN	Optimized PCCN
(1)	$f50\_d, f51\_d, f40\_d, f41\_d, f3\_d, f2\_d\_0 : \{10 - 50\}$	$f5432\_d : \{10 - 50\}$
(6)	$f2\_l\_2 : \{1 - 200\}$	$f2\_l\_2 : \{90 - 130\}$
(5)	$f2\_l\_3 == (f2\_d\_0 - f2\_d\_1) / (2 * tan(f2\_beta/2))$	$f2\_l\_3 == round((f5432\_d - f2\_d\_1) / (2 * tan(f2\_beta/2)), 1)$

Table 3. Comparison between the statistics of the non-optimized and optimized PCCN

	Original PCCN	Optimized PCCN
Number of variables	58	46
Size of the solution space	2.56×10^{37}	2.36×10^{18}
Number of constraints	54	40
Number of binary constraints	33	22
Number of n-ary constraints	21	18

is used for constraint propagation and as a consistency check in forward-checking. In preprocessing, the computational time is analyzed and the extent to which the GAC algorithm can constrain the solution space is investigated. In forward-checking, only the time it takes to find a solution to the problem is measured. The two presented heuristics were applied in both the original and the optimized representation. For transparency, both representations have been published under (Herrmann et al., 2024). The results of the comparison are shown in the table 4.

Table 4. Solving the non-optimized and optimized PCCN

		Original PCCN	Optimized PCCN
Constraint propagation	Runtime [s]	1903.5	139.8
	Size of the solution space	$2.56 * 10^{37}$	$2.36 * 10^{18}$
	Remaining solution space	$7.88 * 10^{31}$	$7.31 * 10^{14}$
Forward-checking	Runtime [s]	7878.8	664.8

The results show that constraint propagation with the GAC algorithm is significantly faster with the optimized PCCN. This is due to the reduced solution space since fewer parameter combinations need to be tested. The GAC algorithm reduces the solution space by six powers of ten in the non-optimized representation and by four powers of ten in the optimized representation. With forward-checking, a solution was found after 7878.8 seconds for the non-optimized representation and after 664.8 seconds for the optimized representation. This corresponds to a runtime reduction of about 92%, demonstrating the efficiency gains through remodeling.

6. Summary and outlook

This paper demonstrates how the Portfolio of Capability Constraint Network (PCCN) enables the modeling and analysis of complex manufacturing networks by formulating a constraint satisfaction problem. However, these models have high problem complexity due to many n-ary constraints and a large solution space, which poses a challenge to solution algorithms. To reduce this problem's complexity, different remodeling approaches from the literature for CSPs were investigated and compared to the specific challenges of solving a PCCN. None of the investigated remodeling approaches represent a practical and efficient problem formulation. Therefore, specific design guidelines and heuristics were developed to reduce the problem complexity of a PCCN. The application of the design guidelines was investigated in the example of the production of a multi-material shaft using tailored forming technology which leads to a significant increase in the efficiency of the solution search.

The results of this paper not only make a scientific contribution by analyzing the characteristics and challenges of solution finding within a PCCN, but also provide practical guidelines for efficient redesign for constraint-based manufacturing networks. While the focus of the use case is on manufacturability analysis, the method and guidelines, together with the PCCN, could be used for all kinds of configuration and optimisation problems in design and manufacturing. In future research, a constraint propagation algorithm will be developed that uses heuristics and a problem-specific approach to establish path consistency in a PCCN, thus reducing the complexity of the problem for subsequent solution methods.

Acknowledgments

The results of this publication have been developed within the framework of the Collaborative Research Center 1153 "Process Chain for the Manufacture of Hybrid High-Performance Components by Tailored Forming" within the subproject C02. The authors would like to thank the German Research Foundation (DFG) for financial and organizational support of the project (project number: 252662854).

References

Akgün, Ö., Gent, I., Jefferson, C., Miguel, I., Nightingale, P. & Salamon, A. Automatic Discovery and Exploitation of Promising Subproblems for Tabulation. *Principles And Practice Of Constraint Programming*. pp. 3-12 (2018)

Albrecht, K. & Anderl, R. Information Model for the Integration of Manufacturing Restrictions into the Algorithm Based Product Development Process. *Procedia CIRP*. 50 pp. 819-824 (2016), https://www.sciencedirect.com/science/article/pii/S2212827116303286, 26th CIRP Design Conference

Anjum, N., Harding, J., Young, R. & Case, K. Manufacturability verification through feature-based ontological product models. *Proceedings Of The Institution Of Mechanical Engineers, Part B: Journal Of Engineering Manufacture*. 226, 1086-1098 (2012,6)

Apt, K. *Principles of constraint programming*. (Cambridge university press, 2003)

Ashby, M. & Cebon, D. Materials selection in mechanical design. *Le Journal De Physique IV*. 3, C7-1 (1993)

Bacchus, F., Chen, X., Van Beek, P. & Walsh, T. Binary vs. non-binary constraints. *Artificial Intelligence*. 140, 1-37 (2002)

Bergenti, F. & Monica, S. Hyper-arc consistency of polynomial constraints over finite domains using the modified Bernstein form. *Annals Of Mathematics And Artificial Intelligence*. 80 pp. 131-151 (2017)

Bessière, C. Non-binary Constraints. *Jaffar (Hg.) 1999 – Principles And Practice Of Constraint*. pp. 24-27

Brailsford, S., Potts, C. & Smith, B. Constraint satisfaction problems: Algorithms and applications. *European Journal Of Operational Research*. 119, 557-581 (1999)

Bulatov, A. A dichotomy theorem for nonuniform CSPs. *2017 IEEE 58th Annual Symposium On Foundations Of Computer Science (FOCS)*. pp. 319-330 (2017)

Clark, D., Frank, J., Gent, I., MacIntyre, E., Tomov, N. & Walsh, T. Local search and the number of solutions. *Principles And Practice Of Constraint Programming—CP96: Second International Conference, CP96 Cambridge, MA, USA, August 19–22, 1996 Proceedings 2*. pp. 119-133 (1996)

Dechter, R. *Constraint processing*. (Morgan Kaufmann, 2003)
Dechter, R. & Pearl, J. Tree clustering for constraint networks. *Artificial Intelligence*. 38, 353-366 (1989)
Domschke, W., Drexl, A., Klein, R. & Scholl, A. *Einführung in Operations Research*. (Springer Verlag, 2015)
Egbunike, O., Purvis, L. & Naim, M. A systematic review of research into the management of manufacturing capabilities. *Production Planning & Control*. 29, 1349-1366 (2018)
Galinier, P. & Hao, J. A general approach for constraint solving by local search. *Journal Of Mathematical Modelling And Algorithms*. 3 pp. 73-88 (2004)
Gembarski, P. Modeling the Portfolio of Capabilities for Product Variant Creation and Assessment. *Proceedings Of The 9th International Conference On Mass Customization And Personalization-Community Of Europe (MCE-CE 2020)*. pp. 77-84 (2020), https://doi.org/10.15488/13513
Gembarski, P. Joining Constraint Satisfaction Problems and Configurable CAD Product Models: A Step-by-Step Implementation Guide. *Algorithms*. 15, 318 (2022), https://doi.org/10.3390/a15090318
Groche, P., Bruder, E. & Gramlich, S. *Manufacturing Integrated Design*. (Springer International Publishing (2017)
Herrmann, K., Brockmöller, T., Gembarski, P., Mozgova, I. & Lachmayer, R. Fertigungsstufen-basiertes Gestaltungsmodell für mechanische Bauteile. *Stuttgarter Symposium Für Produktentwicklung SSP 2021 : Stuttgart, 20. Mai 2021, Wissenschaftliche Konferenz*. pp. 247-258 (2021), https://doi.org/10.15488/14172
Herrmann, K., Bode, B., Wurst, J., Gembarski, P., Mozgova, I. & Lachmayer, R. Quantification of the material-related environmental impact of topology-optimized multi-material components. *Proceedings Of The 33rd Symposium Design For X, DFX 2022*. (2022), https://doi.org/10.35199/dfx2022.01
Herrmann, K., Plappert, S., Gembarski, P. & Lachmayer, R. Process Chain-Oriented Design Evaluation of Multi-Material Components by Knowledge-Based Engineering. *Algorithms*. 16 (2023), https://doi.org/10.3390/a16050247
Herrmann, K., Pusch, F., Plappert, S., Bode, B., Mozgova, I., Gembarski, P. & Lachmayer, R. Prozesskettenorientierte Herstellbarkeitsanalyse von Produkten durch die Nutzung eines Constraint-Satisfaction-Problems. *Stuttgarter Symposium für Produktentwicklung SSP 2023 : Tagungsband zur Konferenz, Stuttgart, 25. Mai 2023*. pp. 79-90 (2023), https://doi.org/10.15488/14221
Herrmann, K. *Portfolio of Capability Constraint Network*. (LUIS, 2023), https://doi.org/10.25835/dx0rtmlv
Regin, J. Generalized arc consistency for global cardinality constraint. *American Association For Artificial Intelligence (AAAI 1996)*. pp. 209-215 (1996)
Jussien, N., Debruyne, R. & Boizumault, P. Maintaining arc-consistency within dynamic backtracking. *International Conference On Principles And Practice Of Constraint Programming*. pp. 249-261 (2000)
Kumar, V. Algorithms for Constraint-Satisfaction Problems: A Survey. *AI Magazine*. 13, 32 (1992)
Lange, K., Kammerer, M., Pöhlandt, K. & Schöck, J. *Fließpressen*. (Springer Berlin Heidelberg, 2008)
Lecoutre, C. Optimization of simple tabular reduction for table constraints. *Principles And Practice Of Constraint Programming: 14th International Conference, CP 2008, Sydney, Australia, September 14-18, 2008. Proceedings 14*. pp. 128-143 (2008)
Lhomme, O. & Régin, J. A fast arc consistency algorithm for n-ary constraints. *AAAI*. pp. 405-410 (2005)
Löffler, S., Liu, K. & Hofstedt, P. The regularization of CSPs for rostering, planning and resource management problems. *Artificial Intelligence Applications And Innovations: 14th IFIP WG 12.5 International Conference, AIAI 2018, Rhodes, Greece, May 25-27, 2018, Proceedings 14*. pp. 209-218 (2018)
Löffler, S. *Optimierung und Regularisierung von Constraint Satisfaction-Problemen (CSPs)*. (BTU Cottbus-Senftenberg, 2022)
Mackworth, A. Consistency in networks of relations. *Artificial Intelligence*. 8, 99-118 (1977)
Marriott, K. & Stuckey, P. *Programming with constraints: an introduction*. (MIT press, 1998)
Nareyek, A. *Constraint-Based Agents*. (Springer Berlin Heidelberg, 2001)
Nellippallil, A., Rangaraj, V., Gautham, B., Singh, A., Allen, J. & Mistree, F. An Inverse, Decision-Based Design Method for Integrated Design Exploration of Materials, Products, and Manufacturing Processes. *Journal Of Mechanical Design*. 140 (2018, 11)
Pahl, G., Beitz, W., Feldhusen, J. & Grote, K. Engineering design: A systematic approach. *Engineering Design: A Systematic Approach*. pp. 1-617 (2007)
Petke, J. Bridging Constraint Satisfaction and Boolean Satisfiability. *Bridging Constraint Satisfaction And Boolean Satisfiability*. pp. 1-113 (2015, 8)
Rossi, F., Petrie, C. & Dhar, V. On the Equivalence of Constraint Satisfaction Problems.. *ECAI*. 90 pp. 550-556 (1990)
Russell, S. & Norvig, P. *Artificial intelligence: A modern approach*. (Prentice-Hall, 2010)
Ruttkay, Z. Constraint satisfaction-a survey. *CWI Quarterly*. 11, 123-162 (1998)
Sabin, D. & Ereuder, E. Understanding and improving the MAC algorithm. *Principles And Practice Of Constraint Programming-CP97: Third International Conference, CP97 Linz, Austria, October 29-November 1, 1997 Proceedings 3*. pp. 167-181 (1997)

Sheveleva, T., Herrmann, K., Wawer, M., Kahra, C., Nürnberger, F., Koepler, O., Mozgova, I., Lachmayer, R. & Auer, S. Ontology-Based Documentation of Quality Assurance Measures Using the Example of a Visual Inspection. *Advances In System-Integrated Intelligence*. pp. 415-424 (2023), https://doi.org/10.15488/13181

Simonis, H. Sudoku as a constraint problem. *CP Workshop On Modeling And Reformulating Constraint Satisfaction Problems*. 12 pp. 13-27 (2005)

Trick, M. A Dynamic Programming Approach for Consistency and Propagation for Knapsack Constraints. *Annals Of Operations Research*. 118, 73-84 (2003)

Tsang, E. Foundations of constraint satisfaction. *Journal Of The Operational Research Society*. 46 (1995)

Walsh, T. SAT v CSP. Principles And Practice Of Constraint Programming - CP 2000. pp. 441-456 (2000)

Wallace, R. “Factor Analytic Studies of CSP Heuristics”. (Springer, Berlin, Heidelberg, 2005) pp. 712-726. https://doi.org/10.1007/11564751_52

Wawer, M., Herrmann, K., Plappert, S., Wurst, J., Gembarski, P., Lachmayer, R. & Others Structuring and Provision of Manufacturing knowledge through the Manufacturing Resource Ontology. *DS 125: Proceedings Of The 34th Symposium Design For X (DFX2023)*. pp. 075-084 (2023), https://doi.org/10.35199/dfx2023.08

Wegener, I. *Complexity theory: exploring the limits of efficient algorithms*. (nSpringer Science & Business Media, 2005)

Zhuk, D. A proof of the CSP dichotomy conjecture. *Journal Of The ACM (JACM)*. 67, 1-78 (2020)