Chapter 20
Applying Constraint Satisfaction Methods in Four-Bar Linkage Design

Gang Chen

Abstract Product design is a very complicated problem. Effective and efficient computerized methods are hence needed to assist human engineers in original product design and design modifications. Constraint satisfaction problem (CSP) provides the insight of problem solving. Many practical algorithms have been developed along with the exploration of CSP's theoretical foundation. This paper applies the general CSP methods into solving product design problems. A crank-rocker mechanism is used as an example to illustrate the proposed ideas.

Keywords Change propagation · Constraint satisfaction · Graph theory · Product design

20.1 Introduction

Mechanical products are usually composed of several components or parts. For a product to realize its designated functions, these parts must be linked in a specified way. Hence, one of the important issues in product geometric design is finding a proper configuration and linking relations for all parts. This task is usually not easy due to the complexity of most mechanical products.

Constraint satisfaction problem (CSP) provides the theoretical foundation and many practical algorithms for finding solutions to complex constraints among multiple variables.

This paper applies CSP methods in four-bar linkage mechanism design. Firstly, theory and basic methods of CSP are briefly introduced; secondly, the four-bar linkage mechanism design is modeled as a CSP with CSP elements specified.

G. Chen (✉)
School of Mechanical Engineering, Tianjin University of Science and Technology,
Tianjin, People's Republic of China
e-mail: chengang@tust.edu.cn

E. Qi et al. (eds.), *The 19th International Conference on Industrial Engineering and Engineering Management*, DOI: 10.1007/978-3-642-38391-5_20,

How to apply constraint propagation and searching methods to find a solution for four-bar linkage mechanism is explained. The purpose is to explore automatic and efficient computerized methods for mechanical or mechatronic product design.

20.2 Constraint Satisfaction Problem

A constraint satisfaction problem is consisted of three elements: a set of variables (V), a set of possible values for each variable (its domain, D), and a set of constraints specified among variables (C). Constraints are rules that impose a limitation on the values that a variable, or a combination of variables, may be assigned simultaneously. A solution to a CSP is an assignment of a value to each variable from its domain such that all constraints are satisfied (Dechter 2003).

Figure 20.1 uses an 8-queen problem as an illustration of CSP. The problem is to place eight queens on an 8 × 8 chessboard satisfying the constraint that no two queens will be a threat to each other (Tsang 1996). One way to model an 8-queen problem as a CSP is as follows: each column in the chessboard is treated as a variable, its domain is all rows in the chessboard, and the constraints are that no two queens should be placed on the same row, column or diagonal.

Constraint satisfaction problems can be represented as constraint graphs in which nodes represent variables while arcs represent constraints between two variables (for binary CSPs). Figure 20.2 is a graph representation of the above-mentioned 8-queen problem.

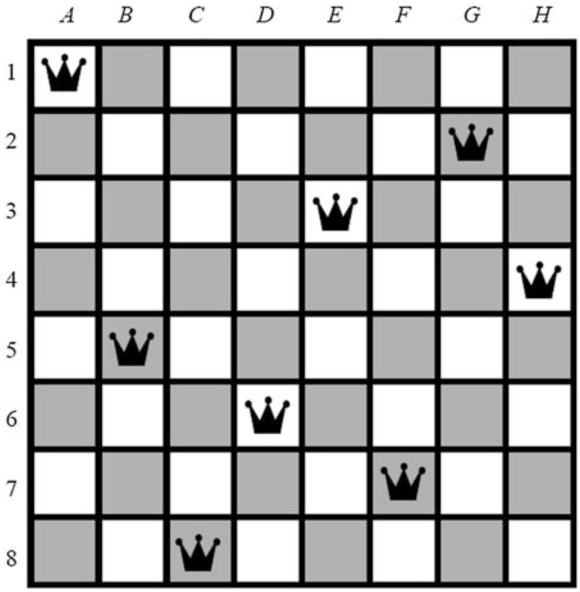


Fig. 20.1 A possible solution to the 8-queen problem (Tsang 1996)

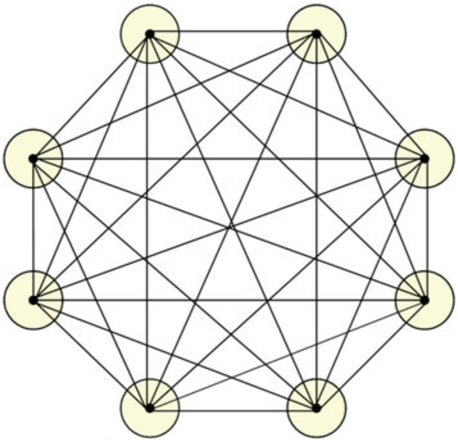


Fig. 20.2 Constraint graph of the 8-queen problem

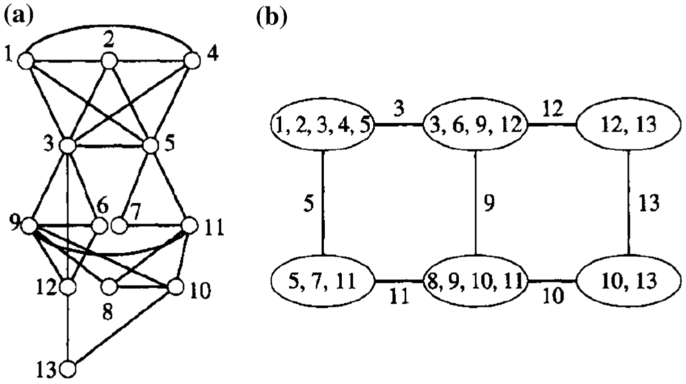


Fig. 20.3 Constraint hypergraph (Dechter 2003)

However, for most practical problems, such as mechanical products design, since most constraints are specified among multiple variables, simple graphs for binary constraints are insufficient. Constraint hypergraph is a more general and suitable representation. Figure 20.3b shows a constraint hypergraph in which each hyperedge (e.g. the circle around variables 1, 2, 3, 4, 5) represents a constraint specified on a set of variables. Arcs between hyperedges represent that two related constraints share one or more variables (labels on arcs).

Two main CSP solving techniques are problem reduction and searching.

(1) *Problem reduction*: Although problem reduction through propagating and processing constraints among related variables can not solve a CSP by itself, however, an original CSP usually can be transformed into an equivalent problem which is easier to be solved.

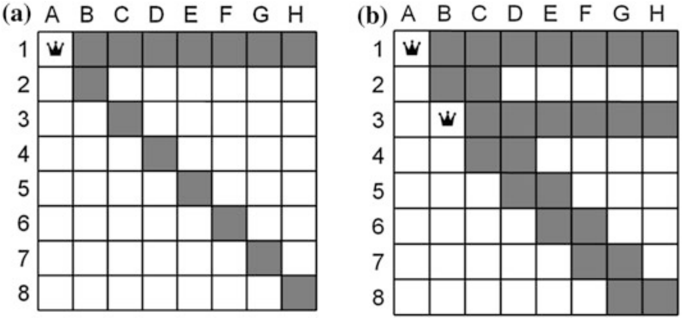


Fig. 20.4 Constraint propagation

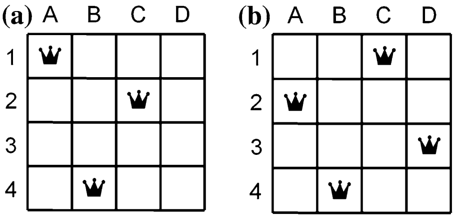
For example, in Fig. 20.4, if the first queen (variable A, which has its value domain as 1–8) is put at cell (1, A), i.e. assigning variable A value 1, then the shaded cells in Fig. 20.4a are excluded from domains of the remaining 7 variables. Figure 20.4b shows further constraint propagation after variable B has been assigned value 3. The problem has become easier to be solved with domain reduction.

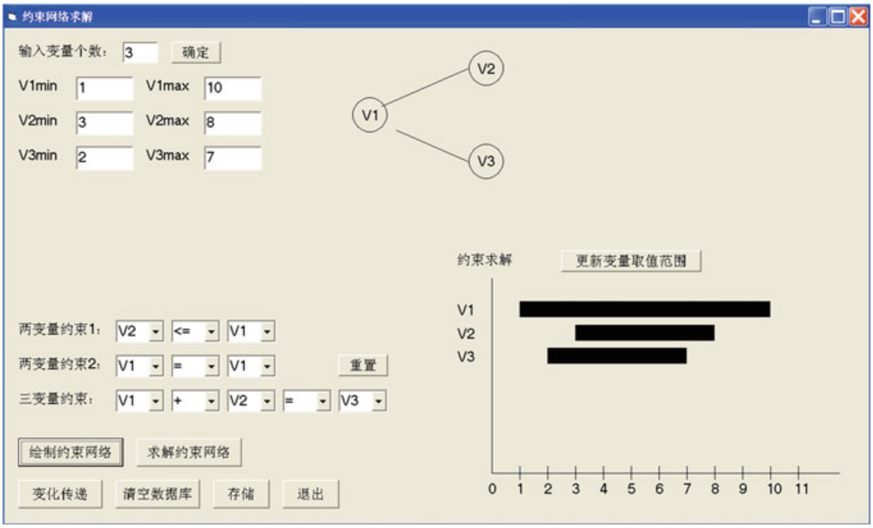
(2) *Search*: This might be the most fundamental technique for solving CSPs. Variables are instantiated one by one. With a partial solution, a new variable is selected and instantiated. If it is impossible to find a feasible value after traversing the new variable's domain, then the searching process is backtracked to select a new value for the previous variable. For a CSP with limited domains, the searching process will carry on until either a solution is found or all the combinations of possible values have been tried and have failed (Tsang 1996). Figure 20.5a shows an inconsistent partial instantiation for the first three variables in a 4-queen problem since no value is feasible for variable D. After backtracking, i.e. assigning a new value to variable A, a solution is found as illustrated in Fig. 20.5b.

A lot of improvements have been made to the simple backtracking algorithm, such as forward checking constraints which involve the most variables or heuristic-guided backtracking instead of chronicle backtracking.

Each constraint may have a weight that indicates its importance. Weight setting represents the priority of a constraint during decision-making. Hard constraints (i.e. constraints that must be satisfied) have higher weights than soft constraints,

Fig. 20.5 Instantiation through searching (Dechter 2003)





约束网络求解

输入变量个数: 3

V1min 1 V1max 10

V2min 3 V2max 8

V3min 2 V3max 7

两变量约束1: V2 <= V1

两变量约束2: V1 = V1

三变量约束: V1 + V2 = V3

Fig. 20.6 A prototype system for constraint solving

which are negotiable or can be relaxed if necessary. For practical problems in real life, a complete solution which satisfies all constraints is usually impossible. It is more realistic to satisfy as many constraints as possible or the most important constraints (with higher weights).

The application of CSP modeling and solving techniques in mechanical design or manufacturing area is rare except for workshop scheduling problem. It might be due to the following reasons:

(1) Realistic problems in mechanical engineering are usually very complex. For example, variables may have different types and abstraction levels; constraints may involve multiple variables and may be represented as complicated mathematical functions.
(2) Value domains of variables in mechanical engineering are usually unlimited and continuous. Traditional searching techniques must be modified.

Figure 20.6 shows a prototype constraint solving system developed using Visual Basic. Interval computing is used to propagate constraints and prune off value domains. Domain discretization based on the precision requirements can be used to transform a continuous problem into a discrete problem.

20.3 Four-Bar Linkage Mechanism Design

The planar four-bar linkage mechanism (crank-rocker mechanism as illustrated in Fig. 20.7) is used here as an example to illustrate how to apply the general CSP solving techniques to mechanical design problems.

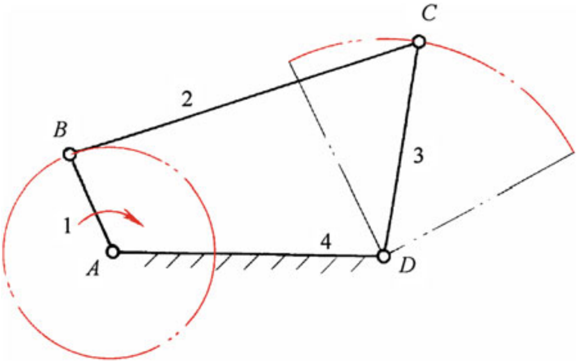


Fig. 20.7 Crank-rocker mechanism (Zhang 2011)

As shown in Fig. 20.7, a planar crank-rocker mechanism has four revolute joints, A, B, C, and D. Link 4 is fixed as the ground link. Link 1 (crank) and link 3 (rocker) are connected to the ground link. Respectively, they are the input and output links of the system.

For a four-bar linkage mechanism to work properly, it must obey several implicitly or explicitly specified rules. For example, let:

$$
\begin{aligned}
S &= length\ of\ the\ shortest\ link\\
L &= length\ of\ the\ longest\ link\\
P &= length\ of\ one\ remaining\ link\\
Q &= length\ of\ the\ other\ remaining\ link\\
Then\ if: &\quad S + L \leq P + Q
\end{aligned}
$$

the Grashof condition indicates that at least one link will be capable of making a full revolution with respect to the ground plane (Norton 1999). Hence, for the mechanism shown in Fig. 20.7 to be a crank-rocker mechanism, lengths of four links must follow the Grashof condition. If lengths of four links are modeled as four variables, then the Grashof condition represented a constraint among variables that the crank-rocker mechanism must satisfied.

One method to design a crank-rocker mechanism is using the specified rocker length, rocker oscillating angle, and the travel velocity-ratio coefficient K as input to synthesize crank length, coupler length, and frame length. The design procedure is described as follows (see Fig. 20.8):

(1) Calculating limitation location angle θ

$$\theta = 180^\circ \frac{K-1}{K+1} \tag{20.1}$$

(2) Selecting a position for joint D, using rocker length L_3 and rocker oscillating angle ψ to determine two extreme rocker positions C_1 and C_2 (as shown in Fig. 20.8);

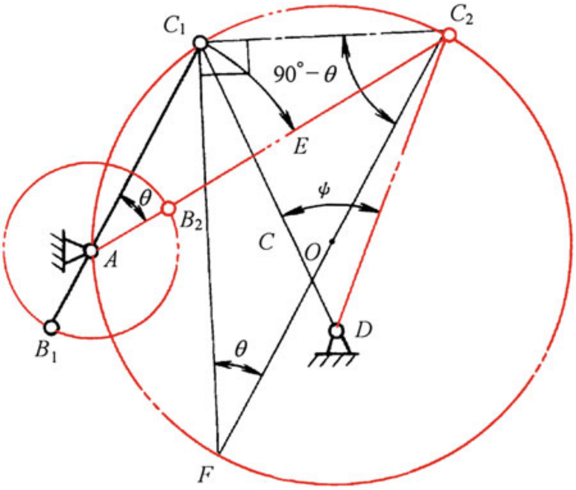


Fig. 20.8 Design procedure of a crank-rocker mechanism (Zhang 2011)

(3) Determining the position of point F as follows: drawing a line C_1F which is perpendicular to line C_1C_2; drawing another line C_2F with the angle $\angle C_1C_2F = 90° - \theta$, θ is the limitation location angle as calculated in step 1;

(4) Making the circumscribed circle of ΔC_1C_2F, another joint A must locate at this circle;

(5) Using the minimum transmission angle γ as the optimization object to finally determine the position of joint A as well as the frame (AD) length L_4;

(6) As shown in Fig. 20.9, the crank length L_1 and the coupler length L_2 can be calculated as follows:

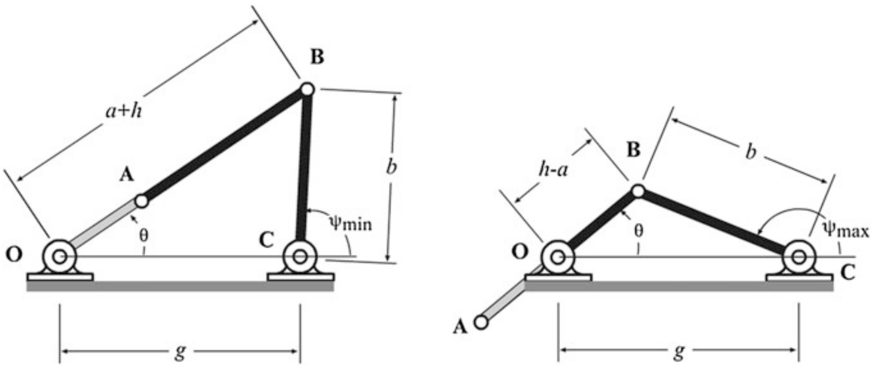


Fig. 20.9 Calculating the crank length and the coupler length (McCarthy and Soh 2010)

$$\begin{aligned} L_1 &= (AC_2 - AC_1)/2 \\ L_2 &= (AC_2 + AC_1)/2 \end{aligned} \quad (20.2)$$

The above-mentioned design procedure can be modeled as a CSP which has the following variables:

(1) Crank length L_1;
(2) Coupler length L_2;
(3) Rocker length L_3;
(4) Frame length L_4;
(5) Travel velocity-ratio coefficient K;
(6) Rocker oscillating angle ψ;
(7) Transmission angle γ;

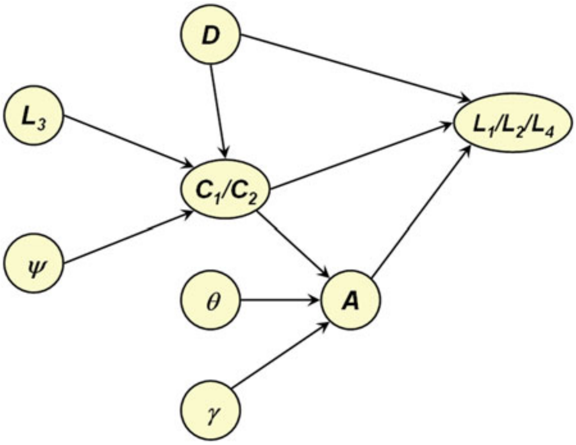
The lower and upper limits for each variable should be specified. The value domain for each variable should be discretized based on the precision requirements. For example, continuous domain (3–8) for rocker length can be transformed into discrete domain (3.0, 3.1,…, 7.9, 8.0).

Some intermediate variables, such as limitation location angleθ, positions of C_1, C_2, F, and A, should also be presented in the constraint graph (see Fig. 20.10).

The Grashof condition, Eqs. (20.1) and (20.2), etc. are modeled as constraints among variables.

The constructed CSP model is shown in Fig. 20.10. Constraint propagation and searching techniques can be used to solve the problem and find a feasible solution.

Fig. 20.10 Constraint graph of crank-rocker mechanism design



20.4 Discussion

A lot of issues need to be addressed before the general CSP methods can be realistically applied into solving mechanical engineering problem (Lottaz et al. 2000; Ribeiro et al. 2008; Ouertani and Gzara 2008; Ermolaeva et al. 2004; Chen et al. 2006; Zhao et al. 2002; Li and Xiao 2004; Jie and Sun 2007; Xu et al. 2002; Li and Xiong 2002). These issues are listed as follows:

(1) Design variables usually have different abstraction levels, how to represent this characteristic using hypergraph;
(2) How to represent complicated mathematical functions as constraints;
(3) How to tackle and manage the complexity of a design problem;
(4) Many disciplines are usually involved in product design, how to represent them in a constraint network;
(5) Product design usually involve several phases in product life cycle, how to model the temporal relations among variables in a constraint graph.

20.5 Conclusion

This paper proposes applying CSP methods to solve mechanical design problems with a crank-rocker mechanism as an example. The purpose is to explore an effective and efficient computerized method for product design. This work is still very primitive. A prototype system is currently under development to illustrate the feasibility of the proposed ideas.

Acknowledgments This research is sponsored by Natural Science Foundation of China (Grant No.51075300), Research Funding from Tianjin University of Science and Technology (Grant No.20090407), and Natural Science Foundation of TianJin (Grant No. 10JCYBJC06800).

References

Chen L, Jin G, Deng L (2006) Research on distributed constraint network solving in multidisciplinary collaborative design. Trans Chinese Soc Agric Mach 37(9):137–140

Dechter R (2003) Constraint processing. Elsevier Science, Amsterdam

Ermolaeva NS, Castro MBG, Kandachar PV (2004) Materials selection for an automotive structure by integrating structural optimization with environmental impact assessment. Mater Des 25:689–698

Jie Y, Sun L (2007) Research of the conflict detection of the collaborative design. J Harbin Univ Sci Technol 12(6):7–9

Li H, Xiao R (2004) Principle of mechanical product design methodology based on constraint network. Chinese J Mech Eng 40(1):121–126

Li T, Xiong G (2002) Multidisciplinary design based on constraint networks. J Tsinghua Univ (Sci Technol) 42(9):1257–1260

Lottaz C, Smith IFC, Robert-Nicoud Y, Faltings BV (2000) Constraint-based support for negotiation in collaborative design. Artif Intell Eng 14:261–280

McCarthy JM, Soh GS (2010) Geometric design of linkages. Springer, New York

Norton RL (1999) Design of machinery. McGraw-Hill, New York

Ouertani MZ, Gzara L (2008) Tracking product specification dependencies in collaborative design for conflict management. Comput Aided Des 40:828–837

Ribeiro I, Pecas P, Silva A, Henriques E (2008) Life cycle engineering methodology applied to material selection, a fender case study. J Clean Prod 16:1887–1899

Tsang E (1996) Foundations of constraint satisfaction. Academic Press Limited, London

Xu Y, Sun S, Pan Y (2002) Research on constraint-based intelligent collaborated design system model. China Mech Eng 13(4):333–336

Zhang C (2011) Theory and design of mechanisms and machines. China Machine Press, Beijing

Zhao H, Tian L, Tong B (2002) Constraint-based conflict detection and negotiation in collaborative design. Comput Integr Manuf Syst 8(11):896–901