Application of CLP for technical objects construction with an exemplary bucket elevation

Grzegorz Bartnik,*, *Andrzej* Marciniak[2], and *Krzysztof* Gołacki[1]

[1]University of Life Sciences Lublin, ul. Głęboka 28, 20-612 Lublin, Poland
[2]University of Economics and Innovation in Lublin, ul. Projektowa 4, 20-209 Lublin, Poland

Abstract. The paper presents the application of the Constraint Logic Programming (CLP) language for designing technical objects. The method was described and illustrated based with the help of on the bucket elevator design. The hierarchical structure of the object, way of system configuration and exemplary sets of fitted components are presented.

1 Introduction

The aim of designing process is a symbol, formal and workable specification of a technical object satisfying specified functional, structural, material, durable resistant, reliable etc. requirements. They can be recorded in the form a finite set of constrains imposed on the quantities characterizing a projected object and depending on the designer`s decision. The quantities, which depend on the designer`s decision, are called decision variables. They can be of different character e.g. numerical, categorical and logical variables as well as elements of specified sets. From the point of view as elements of imposed on the variables, for which the designer imparts values, have the form of predicates or otherwise n-arguments relations.

The complete recording of the construction through specification of constraints imposed on decision variables and recorded in the predicates logic languages becomes a computer program whose execution results in assignment to the decision variables, such values which fulfil a given set of constrains as far as these values exist in the domains corresponding to the variables. The program indicating the construction problem solution recorded in the language e.g. Prolog (Programing Logic) in which the predicates represent construction – exploitation requirements and constrains imposed on the designed object is its sufficiently declarative specification. The development of Prolog Language, adjusting it syntactically and semantically to the description of the problems fulfilling constrains, initiated a new

*Corresponding author: grzegorz.bartnik@up.lublin.pl category of programming languages called Constraint Logic Programming (CLP). The solution satisfying the constrains can be zero or more. In the case there are no such solutions, possibility of constrains attenuation should be taken into consideration. When there are many solutions, one can consider the problem of construction optimization that is searching for the best possible solution after defining one or more criterion functions. The method in which the consideration of requirements imposed on the object and recorded in the CLP language becomes automatically a declarative program determining the variable values being the object of designer's decision. The paper presents the general methodology and gives a specific example of a technical object an with its application in the CLP methodology.

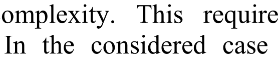
2 Problem with Formulation in the CLP Language

The quantities which acquire values from the designer`s decision are decision variables xi, i = 1 … n denoted by Di, i = 1 … n corresponding to the domain of these decision variables. The domains D1 can be of any type consistent with the semantics of corresponding variables. The constraints imposed on the decision variable values have the form of m – argument predicates where m = 0 or more. The complex two – argument predicates representing the constraining relations between two construction components of the designed object are most frequently applied.

$$r(X,Y){:}{-}C_1(X),\ C_2(Y),\ C_3(X,Y) \qquad (1)$$

where: r, C_1, C_2, C_3 – are the symbols of predicates
X, Y – the variables representing pairs of construction parts
, - the symbol of construction (logical "i"),
: - - the symbol corresponding to logical implication.

The above expression is a typical syntax structure of the Prolog language. The right side is a conjunction of the conditions which must be satisfied in order to fulfill the constraint presented by the left side. Generally, the expression r(X, Y) is interpreted as the statement that the construction element X "fits" the element "Y" or element "X" is compatible with the element "Y". The predicates C1, C_2, C_3 can be also conjunctions of the lower order constraint predicates. The relation "fits" can be also more than the two – argument one. The program declaring the problem under consideration is a set of tasks recorded in accordance with the Prolog language syntax. The program declares the specific solutions of the problem as a set of sentences written with the Prolog language syntax. In order to separate individual constraints, the hierarchical decomposition of the object used as an example in the applied method was made. The constraints can be divided into external and internal ones. The external constraints qualify the requirements for the object as the whole (e.g. output, elevation height etc.) but the internal ones define compatibility between the object components an individual planes of its decomposition. Searching for combination of values of individual variables which satisfy given constraints is a combinatorial problem of exponential calculation complexity. This requires very effective search algorithms including heuristic ones. In the considered case the search method with constraints propagation was applied due to a small complexity of the object used as the designing example. Each constraint was illustrated by means of a suitable search tree. The bipartite graphs are an expressive form of constraints presentation.



where: ○ corresponds to the decision variables and □ represents the constraints which are most often Boolean expressions in the considered case. Constraint Logic

Programming was elaborated in the late eighties of the last century by J. Jaffar and J.L. Lassez. Generally, this programming can be perceived as combination of logic programming, constraints solution and optimization [1, 2 ,3 ,4].

3 Solution of constraints

CLP is a tool which can be used to form constraint solution problems (CSP). Calculation problems accompanying various investigation fields can be modeled and solved as constraint variables solution ones. CSP describes the assumptions of a finite variables number and a finite values number for each variable. The constraint solution problems can be defined as follows:

- The finite set of S variables X_1, X_2, …, $X_{n,}$ is given,
- For each variable – the final set of domains D_1, D_2, …, D_n,
- The set of constrains $C_i(X_{i1}, X_{i2}, …, X_{ik})$ being the subset of the Cartesian product D_{i1} x D_{i2} x … x D_{ik}, which determines the variable values fitting each other in the way defined by the problem in question.

The PSO solution is such on assignment of the values to all variables from their domains that all constraints are met. Additionally, such a solution can satisfy the condition of maximization of and minimization of a given quality index. This is the optimal constraint fulfillment problem (POSO) [5, 6, 7, 8]. In the case of real decision problems the decision space which should be tested by the algorithm proves to be too large. This is the time needed for testing all combinations which can be formed by the algorithm is too long. Therefore, to increase the effectiveness of decision space inspection the constraint logic programming can be applied. It is based on the standard method of recurrences and their modifications combined with constraints propagation. This enables significant constraint of search set and at the same time allows to determine all admissible solutions consistent with the constraints fulfillment problem. Logic programming during problems solving is based on so called active constraints which can include not only materialized variables but also those deprived of value that is immaterialized. There the following domains of CLP can be distinguished:

- logical - the constraints assume only the values: true/false,
- total and real - the constraints are integers and real number,
- linear - all constraints are presented in the form of linear functions,
- finite - where constraints are defined as sets of finite power,
- mixed domains - combination of two or more above mentioned domains.

The finite domain has been the most extensively applied. Some problems in the operation research range are identified with the CLP just in this domain [7, 9, 10, 11, 12].

4 Constraints propagation

Most decision problems in the combinatorial optimization range are practically insolvable for a large number of variables. The search for decision problems solution space by means of complete inspection is ineffective so additional methods of constraint of this set are used. The constraints propagation application is one of the method of decision space constraint in constraint logic programming.

The main assumption of constraints propagation is the application of constraints for reduction of variable domains. Propagation consist in removal from the domain all the variables which do not satisfy the assumed constraints. Each constraint declared in the model is propagated separately whereby every time each constraint propagation brings the values of all other domains up to date. The constraint programming languages can express variables from the range of natural number domains (most often applied) intervals of real numbers, sets and others. However, the constraint propagation itself very rarely gives solution. Therefore it is always combined with distribution and search [1]. In some cases propagation is not a sufficient operation for solving Constraint Satisfaction Problem (CSP), however it increases the process effectiveness (significant acceleration) in order to find an admissible solution. One of the optimization mechanisms used in the CLP is the Branch and Bound (BaB) mechanism. Its aim is to find an optimal solution based on the value of quality index and rejection of all the branches of solution tree which produce the values worse than the current minimum constraint and then to date the minimum constraint bringing up when a better solution than the current one is found [7, 9]. To this end there can be applied one of two methods of decision tree inspection: search deeper or across the decision tree.

5 Exemplary bucket elevator construction

Bucket elevators are used for transport of materials in bulk from one level to another one (higher)) [13]. Their main job of transportation of loose materials to the places of their storage as far as granulation remains at a proper ratio to the sizes of bucket inlet. The transported material does not form a continuous stream but it is placed in suitable containers which makes it different form tape, link, plate or driver conveyors [14, 15, 16, 17, 18]. Figure 1 presents the hierarchical structure of such elevator.

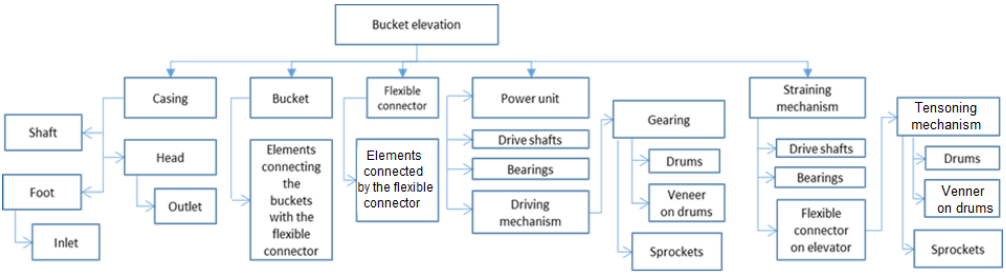
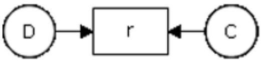


Fig. 1. Hierarchical structure of the object

In order to simplify the choice of individual parts of the bucket elevator dependent on the height of elevation, efficiency and kind of transported material, their review division was made (Tab. 1). The bipartite graph presenting constraints of the elevator kind choice.



D∈{D1, D2} C∈{C1, C2}

r – is interpreted as "fits for" or "is compatible with"

Table 1. List of conditions for the selection of elevator parts and components elevator [15].

Input parameters			
A – elevation height		B – output	C – kind of elevated material
Components			
No.	Name of subassembly/parts	Symbol of subassembly/parts	Selection dependence
1	Kind of elevator	D	C
2	Load-bearing flexible connector	E	A, B, C

3	Bucket	F	B, C, D
4	Power unit	G	A, B
5	Flexible connector drive	H	A, B
6	Flexible connector links	I	A, E
7	Buckets linking	J	E
8	Ratchet	K	B
9	Driving mechanism	L	C, E
10	Straining mechanism	M	C, E
11	Bearings	N	B, C
12	Inlet	O	B, D
13	Outlet	P	B, D
14	Veneer on drums	R	C
15	Casing	S	A, D
A1 – height up to about 25m, A2 – height up to about 80m, B1 – output up about 80t/h, B2 – output up about 300t/h, C1 – nonabrasive material, C2 – abrasive material, **Kind of elevator:** D1 – high-speed (centrifugal), D2 – slow-speed (gravitation) **Load-bearing flexible connector:** E1 – rubber tapes, E2 – plate link chain, E3 – coil chain, E4 – forget chain …			

Table 2. Relation between the kind of bucket elevator and transported material [15].

	D1	D2
C1	1	1
C2	0	1

The relation between the kind of bucked elevator and transported material (Tab. 2) can be presented in the form of search tree (Fig. 2).

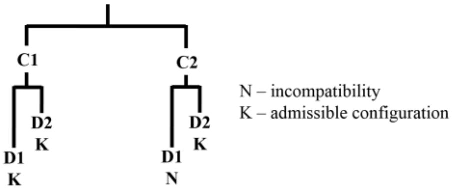


Fig. 2. The search tree using the complete inspection method for the choice of a kind of bucket elevator [15].

Optimal solution of bucket elevator choice depending on the transported material takes construction strength into consideration. Figure 4 presents the search for the best solution based on the complete inspection method.

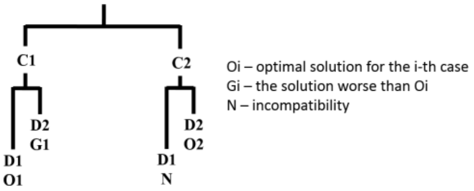

C1
C2
D2
G1
D1
O1
D2
O2
D1
N
Oi – optimal solution for the i-th case
Gi – the solution worse than Oi
N – incompatibility

Fig. 3. The search tree for the optimal choice of the bucket elevator based on the strength criterion [15].

As follows from Fig. 3 in the case of nonabrasive materials transport the best solution is the use height–speed elevator but for abrasive materials transport the slow–speed elevator proves to have better strength.

The bipartite graphs presenting constrains in the choice of the kinds of carrying connector de-pending on the height of transported material (A), output (B) and the kind of transported material (C).

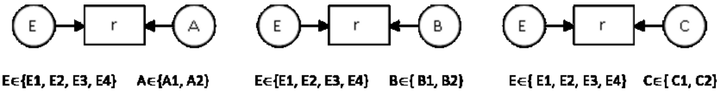


Table 3. Dependence of flexible connector choice [15].

	E1	E2	E3	E4		E1	E2	E3	E4		E1	E2	E3	E4
A1	1	1	1	1	B1	1	1	1	0	C1	1	1	0	0
A2	1	0	1	0	B2	1	1	1	1	C2	1	0	1	1

Table 3 presents the specification of kinds of flexible connectors used in bucket elevators. However, its graphic image is Figure 4 showing the tree of complete search for flexible connector depending on the height of elevator lifting, output and kind of transported material. The search is often made for optimal solution whose price is the lowest. This solution can be obtained by means of the Branch and Bound (BaB) method [14]. Table 4 presents the result of search for optimal solutions.

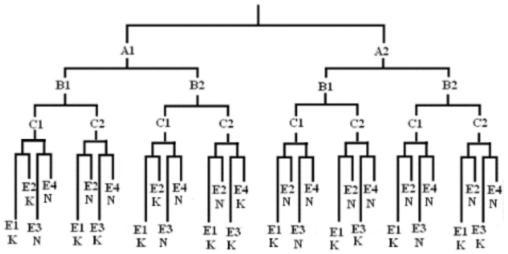


Fig. 4. The search tree using the complete inspection method for the choice of flexible connector [10].

The BaB method can be used for the choice of a suitable connector depending on the kind of bucket elevator (Fig. 5).

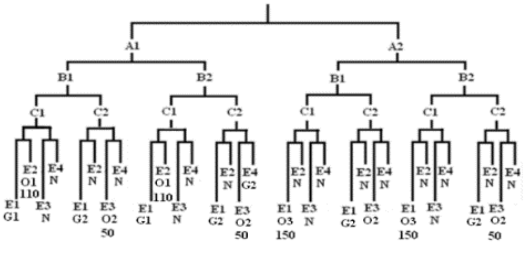


Fig. 5. The search tree using the BaB method for the optimal choice of flexible connector [15].

Table 4. The sets of fitted components [15].

Number of set	**Conditions of elevator work**	**Optimal set (WEH – with elevation height control, OEH – without elevation height control)**
I.	A1, B1, C1	Z1 = {D1, E2, F9, G1, H1, I9 (ZRP), I8 (BRP), J3, K2, L3, M3, N1, O1, O4, P1, P3, S1, S4, S5}

II.	A2, B1, B1	Z2 = {D1, E1, F9, G1, H1, I6 (ZRP), I2 (BRP), J1, K2, L1, M1, N1, O1, O4, P1, P3, R1, S1, S3, S5}
III.	A1, B2, C1	Z3 = {D1, E2, F9, G1, H1, I9 (ZRP), I8 (BRP), J3, K2, L3, M3, N2, O2, O4, P2, P3, S1, S4, S5}
IV.	A1, B1, C2	Z4 = {D2, E3, F9, G1, H1, I12 (ZRP), I12 (BRP), J5, K2, L6, M3, N2, O1, O3, P1, P4, S2, S4, S6}
V.	A2, B2, C1	Z5 = {D1, E1, F9, G1, H1, I6 (ZRP), I2 (BRP), J1, K2, L1, M1, N2, O2, O4, P2, P3, R1, S1, S3, S5}
VI.	A2, B1, C2	Z6 = {D2, E3, F9, G1, H1, I12 (ZRP), I12 (BRP), J5, K2, L6, M3, N2, O1, O3, P1, P4, S2, S3, S6}
VII.	A1, B2, C2	Z7 = {D2, E3, F9, G1, H1, I12 (ZRP), I12 (BRP), J5, K2, L6, M2, N3, O2, O3, P2, P4, S2, S4, S6}
VIII.	A2, B2, C2	Z7 = {D2, E3, F9, G1, H1, I12 (ZRP), I12 (BRP), J5, K2, L6, M2, N3, O2, O3, S2, S3, S6}

The process knowledge in the informal form can be transformed to the form of codes by means of such conceptual schemes as bipartite graphs (Fig. 6). From the point of view of logics, a language of bipartite graphs passes expressiveness of predicates logic of the first order which seems to be sufficient for formal description of the structures of technical objects.

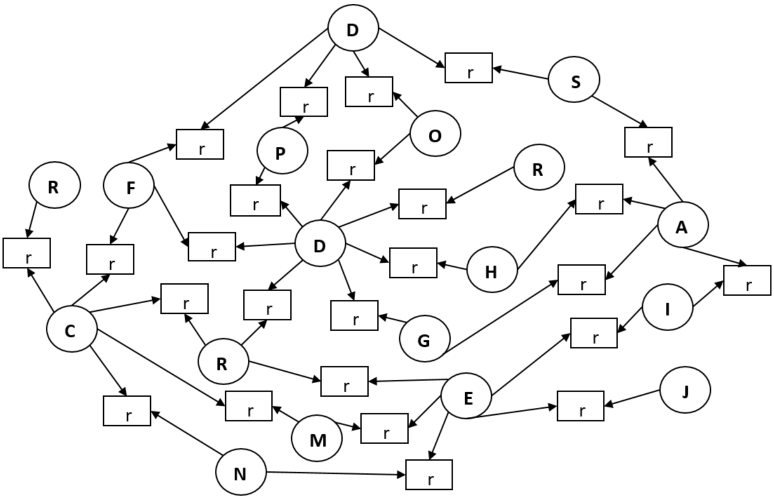


Fig. 6. The object structure presented by means of bipartite graphs comparison [15].

6 Conclusions

The paper presents the application of programming methods in the constraint logic programming (CLP) language for designing technical objects. It describes the designing method illustrated by the example of the bucket elevator projection. The hierarchical structure of the object, way of system configuration as well as the sets of fitted components one included. The presented way of projection is exemplary and simplified. In fact the number of constraints imposed on the projected technical objects is much higher (e.g. concerning safety, environmental protection etc.). Due to the calculation complexity domains of individual decision variables should have a finite and not very large number of values. This indicates that the described projection method is suitable for the approach used for the configuration of projected object from the standardized and earlier projected ready components of known characteristics taken into consideration in specification and control of compatibility constraints. This way of projection object conceptualization is of universal character. It can refer to projection of technical objects, construction materials, bio- and nano - technological products as well as food produce. Formalization of the above presented conceptualization is presented and exemplary illustrated.

7 References

1. http://www-users.mat.umk.pl/~fly/materialy/pl/referaty/CLP_rosi.pdf (2018)
2. J. Jaffar, M. J. Maher, *Constraint Logic Programming: A Survey*, Journal ofLogic Programming, **19-20**, 503-581 (1994)
3. T. Tatoń, Wybrane problemy aukcji i przetargów kombinatorycznych. http://docpl ayer.pl /37987537-Wybrane-problemy-aukcji-i-przetargow-kombinatorycznych.html (2017)
4. H.P. Williams, *Logic and Integer Programming*, Springer (2009)
5. K. Apt, M. Wallace, *Constraint Logic Programming using Eclipse*, Cambridge University Press, (2006)
6. R. Bart́ak, *On-Line Guide to Prolog Programming*, http://kti.mff.cuni.cz/ bartak/prolog/, (2010)
7. J. Hansen, *Constraint Programming versus Mathematical Programming*, Orbit, 3 (2003)
8. A. Niederliński, *Programowanie w logice z ograniczeniami*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, (2010)
9. http://www.hakank.org/ECLiPSe/, (2018)
10. Janczura K., Gabiga T. *Wprowadzenie do Programowania Logicznego z Ograniczeniami z wykorzystaniem ECLiPSe*. PWSZ Tarnów (2009)
11. Ł. Mazur, *Paradygmat programowania CLP. Metody rozwiązywania trudnych problemów kombinatorycznych*. http://software.com.pl (2012)
12. T. Żurek, *Metody sztucznej inteligencji*, Wydawnictwo UMCS, (2011)
13. Goździecki M., Świątkiewicz H. *Przenośniki*, Wydawnictwo Naukowo – Techniczne, Warszawa 1979.
14. L. W, Kurmaz, O. L. Kurmaz, *Projektowanie węzłów i części maszyn podręcznik* (2007)
15. A. Lipczak, *Koncepcja konstrukcji segmentowego przenośnika kubełkowego*, AR Lublin (2012)
16. M. Markowski, *Przenośniki*, Wydawnictwo Politechniki Łódzkiej, I, edition III (1995)
17. M. Markowski, *Przenośniki*, Wydawnictwo Politechniki Łódzkiej, II, edition II (1986)
18. R. Raczyk, *Środki transportu bliskiego i magazynowania*, Wydawnictwo Politechniki Poznańskiej, (2009)