Declarative Model of Competences Assessment Robust to Personnel Absence

Eryk Szwarc[1](✉), Grzegorz Bocewicz[1], Irena Bach-Dąbrowska[2], and Zbigniew Banaszak[1]

[1] Faculty of Electronics and Computer Science,
Koszalin University of Technology, Sniadeckich 2, 75-453 Koszalin, Poland
eryk.szwarc@tu.koszalin.pl
[2] Finance and Management Faculty,
WSB University of Gdansk, al. Grunwaldzka 238A, 80-266 Gdansk, Poland
ibach@wsb.gda.pl

Abstract. Workers' absences are a common disruption to the provision of tasks. They make it necessary to modify task assignment, which amounts to finding suitable substitutions. Sometimes it happens that the competences of the available workers, with given constraints, e.g. hour limits, are not sufficient to find an admissible assignment modification. Therefore, it is desirable to develop a so-called robust personnel competence structure. In the case of workers' absences, a robust competence structure is a structure that allows one to find a modification of the assignment under every possible scenario of disruption. In other words, the following question is considered: does there exist, and if so, what is the competence structure robust to the disruption caused by an unexpected personnel absence? The number of potential solution variants is related to the number of competences that can be improved. For each variant, there could exist many cases of worker absence (absence of one worker, two workers, etc.). Moreover, for these, there could also exist many assignment modification variants. This an NP-hard problem. In the context of the scale of problems encountered in practice, searching for solutions using well-known algorithms (based on an exhaustive search) is a time-consuming process which does not guarantee that an admissible solution will be found. In the present study, a sufficient conditions are proposed, the fulfilment of which guarantees the existence of a non-empty set of admissible solutions. Declarative modelling and computer implementation in the form of constraint logic programming (CLP) have been applied. The potential of the proposed solution is illustrated with examples.

Keywords: Competence assessment · Declarative model · Robustness · Personnel absence · Worker assignment

1 Introduction

Task assignment is an important component of the complex scheduling problem. In a general case, the Task Assignment Problem (TAP) involves assigning resources (workers) to tasks so as to satisfy the constraints related to the personnel hour limits,

R. Damaševičius and G. Vasiljevienė (Eds.): ICIST 2019, CCIS 1078, pp. 12–23, 2019.
https://doi.org/10.1007/978-3-030-30275-7_2

workload, etc. [7, 8, 14]. An obvious criterion in assigning workers to tasks due to their competences, understood as a set of knowledge, experience and skills – elements that allow particular workers to realize particular tasks [4, 9, 11, 19]. The individual competences of specific employees make up the competence structure of the available personnel. This structure determines, in a natural manner, the possibility of providing a specific group of tasks. In the literature, there exist models and algorithms which allow one to answer typical questions connected with the assessment of the personnel's potential and estimation of the costs of assignment, i.e. issues in the area of analysis of competence structures [1, 5, 6, 10, 13, 15], for example: Does the given personnel with the given competence structure make it possible to provide tasks under the assumed constraints?

There are also models and algorithms which allow to answer questions connected with the possibility of strengthening the potential of the existing personnel, i.e. issues in the area of synthesis of competence structures (the few studies in this area include [17, 18]), for example: Does there exist (an improved) competence structure of the given personnel such that the worker assignment satisfies the constraints assumed?

In practice, every worker assignment plan is exposed to a defined set of disruptions, such as worker absences, changes in task numbers, etc. In this present article, we focus only on worker absences which may require modifications to the worker assignment, i.e finding suitable substitutions for absent employees. An answer to the following question is sought: Does the given competence structure of the available personnel allow to modify the assignment in the case of worker absence?

Sometimes, it happens that the given competence structure is not sufficient to find an admissible assignment modification. This type of structure is said to be non-robust to a given disruption. Cases like that lead to the formulation of a new generation of human resources management problems (omitted in the literature), which focus on the possibility of searching for a data structure (a competence structure in this case) which allows to assign resources so that specific expectations (e.g. robustness of the assignment plan to worker absence) are met [16]. In other words, an answer to the following question is sought: Does there exist (an improved) competence structure of the available personnel such that an admissible assignment modification can be made in the case of worker absence?

The number of alternative competence structures which guarantee an admissible assignment modification is related to the number of competences that can be improved. For each variant of an improved competence structure, there could exist many cases of disruption. Moreover, for each such case, there could also exist many assignment modification variants. The process of searching for a solution to this problem is illustrated using a simplified example (see Sect. 2). It shows that in the problem under consideration, the number of solutions increases exponentially. In the context of the scale of problems encountered in practice, searching for solutions using well-known algorithms (based on exhaustive search) is a time-consuming process that does not at all guarantee that an admissible solution will be found. The purpose of the present work is to propose the sufficient conditions the fulfilment of which will guarantee a non-empty set of admissible solutions.

The remainder of this paper is organized as follows: Sect. 2 provides an illustrative example of the problem of planning competence robust to unexpected personnel absence. Section 3 presents a declarative model of the problem considered. Section 4 shows the constraint satisfaction problem as an effective solving method. The last part (Sect. 5) contains conclusions and indicates the main directions of future research.

2 Motivation Example

Let us consider the following simplified example of robustness assessment of a competence structure and planning a competence structure that will be robust to unexpected worker absence. The example shows that in general the space of solutions is large, which makes the problem considered non-trivial.

Enterprise realize project, whose success is conditioned by the performing 10 tasks $o_{i,j}$ (Fig. 1a). Tasks are grouped in terms of the competences required for their realization: $O_1 = \{o_{1,1}\}, O_2 = \{o_{2,1}, o_{2,2}, o_{2,3}\}, O_3 = \{o_{3,1}, o_{3,2}\}, O_4 = \{o_{4,1}, o_{4,2}, o_{4,3}, o_{4,4}\}$. For example this means that tasks $o_{2,1}, o_{2,2}, o_{2,3}$ require the same competence hereinafter referred as O_2, ($o_{i,j}$ – means j-th task requiring i-th competence). It is assumed that each task takes 30 units of time (u.t.) and the technological order of performing tasks is known (Fig. 1a). For example, the beginning of the task $o_{2,1}$ is conditioned by the completion of $o_{4,1}$, beginning of the task $o_{2,2}$ is conditioned by the completion of $o_{4,2}$ i $o_{4,3}$, etc.

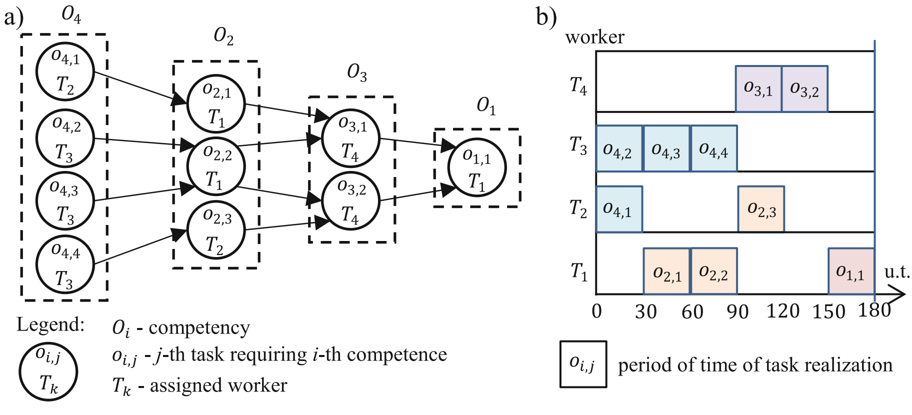

Fig. 1. Structure of considered project (a), schedule determined by assumed assignment (b)

Given are M workers $T_k (k = 1 \ldots M)$. Each worker T_k is characterised by a binary set of indicators $g_k = (g_{k,1}, \ldots, g_{k,i}, \ldots, g_{k,N})$:

- $g_{k,i} = 1$ indicates that the worker T_k has the competency O_i,
- $g_{k,i} = 0$ indicates that the worker T_k has no competency O_i.

For example, the notation $g_1 = (1, 1, 0, 0)$ means that worker T_1 has competences O_1 and O_2, and has no competences O_3 and O_4. Together, the indicators for all worker make up the competence structure of the worker team: $G = (g_{k,i}|k = 1...M; i = 1...N)$, where $g_{k,i} \in \{0, 1\}$. In the example, let us consider the following competence structure G:

$$G = \overset{\text{Competences } O_i}{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}} \text{Workers } T_k$$

Each worker T_k has a limited number of hours to realize (minimum s_k and maximum z_k). An example of limitations are shown in Table 1.

Table 1. Minimum and maximum workers hour limits

	s_k	z_k
T_1	90	120
T_2	60	120
T_3	90	120
T_4	60	90

It is known that there exists a task assignment (number of tasks of given competence group O_i assigned to the k-th worker): $X = (x_{k,i}|k = 1...M; i = 1...N)$, where $x_{k,i} \in \mathbb{N}$, which satisfies the assumed limits s_k and z_k. An example of a variant of the assignment is given below:

$$X = \overset{\text{Competences } O_i}{\begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 0 \end{bmatrix}} \text{Workers } T_k$$

Moreover, given is variant of project diagram with assigned employees (Fig. 1a). It shows which task $(o_{i,j})$ is assigned to which employee. For example, task $o_{4,1}$ is assigned to worker T_2, tasks $o_{4,2}$, $o_{4,3}$, $o_{4,4}$ are assigned to worker T_3, etc.

Project schedule assumes the possibility of its completion in 180 u.t. (units of time) (Fig. 1b).

As announced in the Introduction, an unexpected worker absence will be considered. An unexpected absence means that any worker and any number of workers could be absent during the of assignment X. Of course, we do not know which workers will be absent. All cases (the absence of one worker, two workers, etc.) must be considered. In general, the number of all disruption cases is $PC = \sum_{a=1}^{M} C_M^a$, where M – the number of workers. In the example considered, $M = 4$, and so $PC = 15$:

(a) 4 scenarios of one worker's absence,
(b) 6 scenarios of two workers' absence,
(c) 4 scenarios of three workers' absence,
(d) 1 scenario of four workers' absence.

To assess the robustness of the competence structure to worker absence, one has to answer the following question: does the given competence structure G allow to make an admissible modification to worker assignment X under each absence scenario? An admissible modification should be understood as a worker substitution for tasks which are assigned to the absent worker. For example, the absence of one worker requires the following modifications to assignment X:

- if the absent worker is T_1, then substitute workers are needed for tasks: $o_{1,1}$, $o_{2,1}$, $o_{2,2}$.
- if the absent worker is T_2, then substitute workers are needed for tasks: $o_{2,3}$, $o_{4,1}$.
- etc.

First, let us consider case (a) and all the scenarios of one worker's absence – see Fig. 2. It can be seen that for two out of the four possible scenarios (the absence of T_3 and T_4), no admissible modification of worker assignment X can be made.

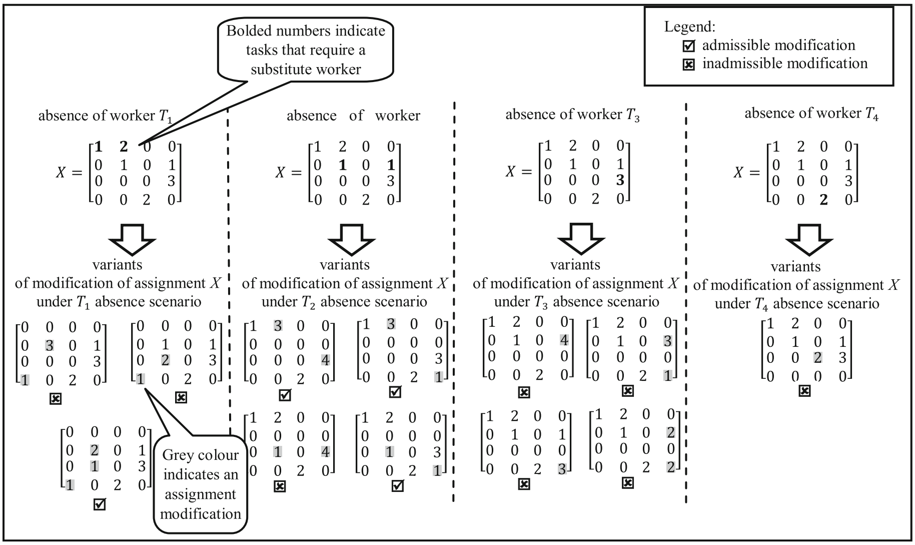
Bolded numbers indicate tasks that require a substitute worker
Legend:
☑ admissible modification
☒ inadmissible modification
absence of worker T_1
absence of worker
absence of worker T_3
absence of worker T_4
$X =$
variants of modification of assignment X under T_1 absence scenario
variants of modification of assignment X under T_2 absence scenario
variants of modification of assignment X under T_3 absence scenario
variants of modification of assignment X under T_4 absence scenario
Grey colour indicates an assignment modification

Fig. 2. Variants of modification of worker assignment X for all scenarios of a one worker absence

An important observation for each scenario of cases (b) and (c) is that the total number of working hours (300) is greater than the sum of the maximum number of hours z_k for each worker $\left(\sum_{k=1}^{M} z_k\right)$. Thus, again, no admissible modifications can be made to these cases.

Lastly, it is obvious that case (d) is a special situation when it is not possible to make any admissible modifications.

In conclusion, the answer is: the given competence structure G guarantees an admissible modification of worker assignment X under two out of the 15 worker absence scenarios. In other words, competence structure G is not robust to unexpected worker absence. Therefore, a reverse question must be asked: Does there exists a competence structure G such that an admissible modification can be made to worker assignment X under every possible absence scenario? Or, put differently, which competency $g_{k,i}$ of which worker T_k should be improved to ensure an admissible modification of assignment X? Competency improvement is understood as changing the value of competency in structure G from “0” to “1”. Thus, all the possible variants of competence structures can be represented by the general formula $2^{(M \cdot N) - CP} - 1$, where CP is the number of competences with value “1” in structure G (i.e. competences which cannot be improved). In our example, $M = 4$, $N = 4$, $CP = 10$, which gives $2^{(4 \cdot 4) - 10} - 1 = 63$ variants.

It is obvious that in the example considered whatever the structure is, it would not be robust in cases (b), (c) and (d). It is only possible to find a competence structure robust to the scenarios of case (a).

It is necessary to use an exhaustive search to find an admissible solution. A tree of competence structure variants is shown in Fig. 3. Some of these variants are marked as allowing an admissible modification of X under all the scenarios of one worker’s absence. The answer to the question above, then, is that there exists a competence structure G such that an admissible modification of worker assignment X can be made under every possible scenario of one worker’s absence. To prove it, let us choose competence structure indicated by blue color on Fig. 3. Different schedules under every possible scenario are illustrated on Fig. 4.

However, in special cases, none of the competence structure variants guarantee an admissible modification of worker assignment X. This is especially important in real-world-sized problems, for example: $M = 10$, $N = 30$ and CP around 200, where there are $2^{(10 \cdot 30) - 200} - 1 = 2^{100} - 1$ possible competence structure variants. Because in such cases calculations are highly time consuming, and still do not guarantee that an admissible solution will be found, the following question seems worth considering: does there exist a sufficient condition such that an admissible solution is guaranteed to exist? To find an answer to this question, a competence structure with values “1” only should be considered. Let us call this structure a full-competence structure. Thus, the sufficient condition can be formulated as follows: IF for the full competence structure it is possible to make a modification to worker assignment X under each absence scenario, THEN at least one admissible solution exists. Thus, only this one variant of the competence structure should be checked for admissible solutions.

In general, the problem considered can be formulated using:

- sets of tasks and workers and the parameters that define their quantitative measures,
- decision variables that determine worker assignments and their competence structure,
- and constraints which link the sets and the decision variables.

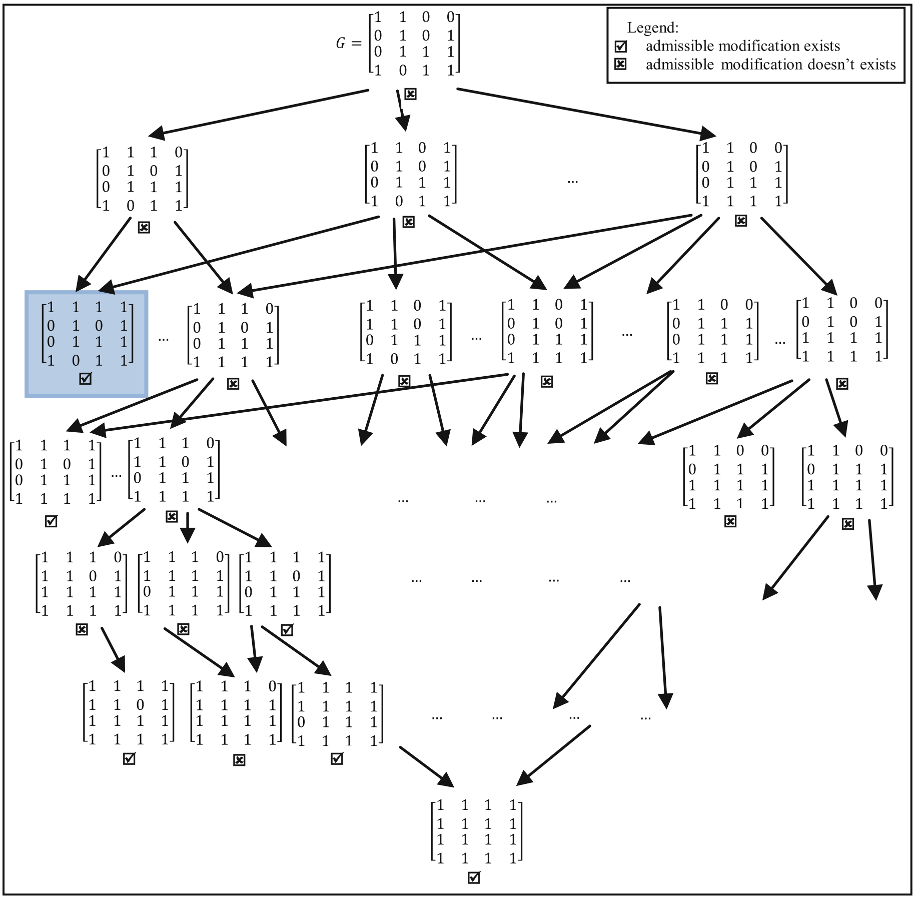
Legend:
☑ admissible modification exists
☒ admissible modification doesn't exists
$G = \begin{bmatrix}1&1&0&0\\0&1&0&1\\0&1&1&1\\1&0&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&1&0\\0&1&0&1\\0&1&1&1\\1&0&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&0&1\\0&1&0&1\\0&1&1&1\\1&0&1&1\end{bmatrix}$ ☒
...
$\begin{bmatrix}1&1&0&0\\0&1&0&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&1&1\\0&1&0&1\\0&1&1&1\\1&0&1&1\end{bmatrix}$ ☑
...
$\begin{bmatrix}1&1&1&0\\0&1&0&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&0&1\\1&1&0&1\\0&1&1&1\\1&0&1&1\end{bmatrix}$ ☒
...
$\begin{bmatrix}1&1&0&1\\0&1&0&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
...
$\begin{bmatrix}1&1&0&0\\0&1&1&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
...
$\begin{bmatrix}1&1&0&0\\0&1&0&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&1&1\\0&1&0&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☑
...
$\begin{bmatrix}1&1&1&0\\1&1&0&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
... ...
$\begin{bmatrix}1&1&0&0\\0&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&0&0\\0&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&1&0\\1&1&0&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&1&0\\1&1&1&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&1&1\\1&1&0&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☑
... ...
$\begin{bmatrix}1&1&1&1\\1&1&0&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ ☑
$\begin{bmatrix}1&1&1&0\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ ☒
$\begin{bmatrix}1&1&1&1\\1&1&1&1\\0&1&1&1\\1&1&1&1\end{bmatrix}$ ☑
... ...
$\begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ ☑

Fig. 3. Tree of competence structure variants (Color figure online)

Computing environments that are well adapted to solving this type of problems are declarative programming environments [2, 3]. They reflect the structure of the constraints of a task in a natural way by modelling the task as a so-called constraint satisfaction problem (CSP) [12].

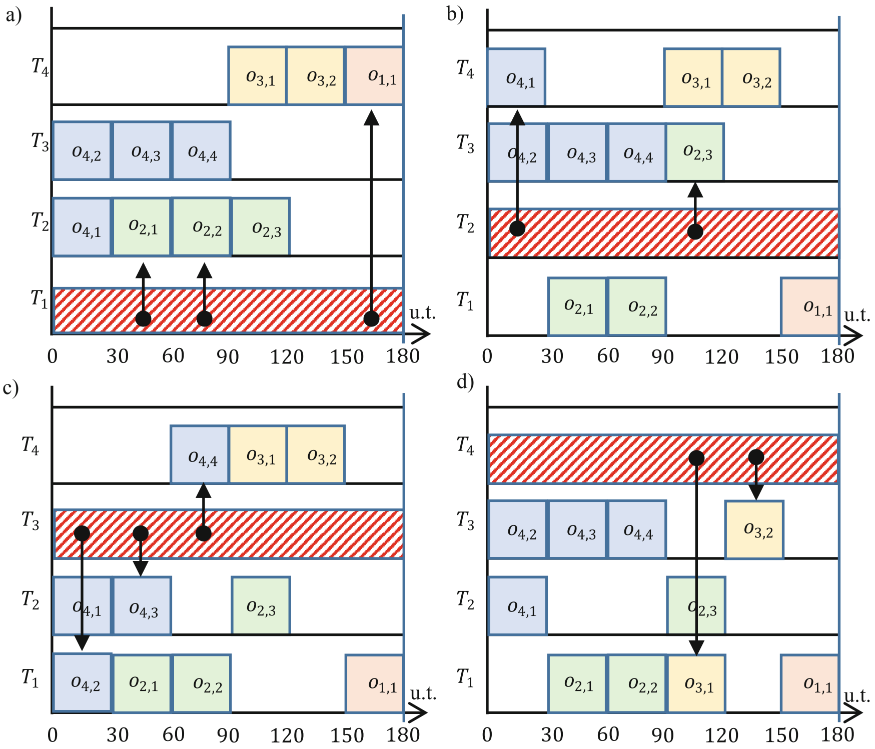

Fig. 4. Schedule under T_1 absence (a); T_2 absence (b); T_3 absence (c); T_4 absence (d)

3 Declarative Model

The following declarative model is proposed:

Sets:
O_i: competences, indexed by $i = 1, \ldots, N$,
T_k: workers, indexed by $k = 1, \ldots, M$.

Parameters:
s_k: minimum number of hours of the k-th worker ($s_k \in \mathbb{N}$),
z_k: maximum number of hours of the k-th worker ($z_k \in \mathbb{N}$).

Decision variables:
G: competence structure, defined as $G = (g_{k,i} | k = 1\ldots M; i = 1\ldots N)$, where $g_{k,i}$ stands for a worker's competences to realize a given task; $g_{k,i} \in \{0, 1\}$, $g_{k,i} = 0$ indicates the k-th worker has no competence O_i, $g_{k,i} = 1$ indicates the k-th worker has competence O_i.

G^j: competence structure obtained as a result of the absence of the j-th worker, $G^j = (g^j_{k,i}|k = 1\ldots(M-1); i = 1\ldots N)$

X^j: task assignment for structure G^j (in the situation of the j-th worker's absence), defined as $X^j = (x^j_{k,i}|k = 1\ldots(M-1); i = 1\ldots N)$, where $x^j_{k,i} \in \mathbb{N}$ means the number of tasks $o_{i,j}$ assigned to the k-th worker in structure G^j.

Constraint related to competences:

(a) tasks can be assigned to the workers with appropriate competences:

$$x^j_{k,i} = 0, \text{ where } g^j_{k,i} = 0 \tag{1}$$

Constraints related to the number of working hours:

(b) The number of hours assigned to T_k should be greater than or equal to the minimum number of hours given to the k-th worker:

$$\sum\nolimits_{\mathrm{i}=1}^{N} x^j_{k,i} \cdot l_i \geq s_k, \text{ where } k = 1, \ldots, M \tag{2}$$

(c) The number of hours assigned to T_k should be less than or equal to the maximum number of hours given to the k-th worker:

$$\sum\nolimits_{\mathrm{i}=1}^{N} x^j_{k,i} \cdot l_i \leq z_k, \text{ where } k = 1, \ldots, M \tag{3}$$

Constraint related to absence:

(d) Construction of competency structures corresponding to the situation of the j-th workers' absence:

$$g^j_{k,i} = \begin{cases} g_{k,i} & \textit{when } k < j \\ g_{(k+1),i} & \textit{when } k \geq j \end{cases} \tag{4}$$

Task assignment X^j which satisfies all the constraints (1)–(4) is known as the admissible worker assignment in situation of the j-th workers' absence.

In this model, the following questions can be formulated:

(a) Does the given competence structure G guarantee an admissible worker assignment?

(b) Does there exist, and if so, what is the competence structure G that guarantees the existence of an admissible worker assignment?

(c) For the given competence structure G, does there exist a modification to the given worker assignment that can be made in the case of worker absence?

(d) Does there exist, and if so, what is the competence structure G which guarantees that a modification can be made to the worker assignment in the case of worker absence?

We are looking for a solution where the values of all the variables satisfy all the constraints (the first solution found, or all of them). This is the well-known constraint satisfaction problem (CSP) [3].

4 Method

The solution to a CSP is obtained as a result of a systematic search of all the possible assignments of values to decision variables. The search methods can be classified according to whether the entire space of all the possible assignments is searched (an exhaustive search) or whether we search only a part of this space (a combinatorial search). The exponential complexity of an exhaustive search, even assuming that 90% of potential solutions are rejected because they do not satisfy the constraints, make them impractical to use in most everyday situations. Thus, combinatorial search methods remain the only alternative. These approaches use various heuristics to reduce the number of necessary searches in a substitution tree (heuristics that constitute a generalization of the sequence of experiments carried out earlier). Both exhaustive and combinatorial search methods use the same iterative scheme: constraint propagation and variable distribution. In other words, the variables declared in the script of a program, along with their domains and the constraints that link decision variables (relations, Boolean and algebraic expressions, etc.) are processed by the same combination of mechanisms of propagation and distribution [3, 12].

The values of the variables that do not satisfy the constraints are removed from their domains during constraint propagation. In most cases, however, the final result cannot be reached by the propagation of constraints alone. It is necessary to introduce a distribution of variables together with searching. The distribution of variables consists in introducing an additional constraint (this is often accomplished by assigning a value to one of the decision variables) and checking its compatibility (consistency) by propagating the constraints. This may result in one of the three possible outcomes:

- a solution is found (each variable has one value from its domain),
- the domains of some of the variables are narrowed down, however, no solution is found yet. This means the distribution of another variable is necessary,
- the additional constraint is incompatible with the remaining constraints; backtracking is performed and the current value of the variable is removed from the domain.

In this iterative search process, a search tree is generated in which each node corresponds to a certain state of a variable. These mechanisms were implemented in the Matlab programming environment. The tool developed in this way was used in computational experiments to assess the time of the determination of a competence structure robust to workers' absences, for a variable number of workers (5–15) and a variable number of tasks (16–32). The results are shown in Table 2. As it can be easily observed, for those assignments that involve fewer than 10 workers and 32 tasks, the time needed to determine a robust competence structure does not exceed 1000 s. Moreover, the competence structures obtained require a minimum number of changes.

Our future work will focus on the implementation of the model proposed in commercial optimization solvers, such as IBM ILOG CPLEX Optimizer, Gurobi Optimizer, Oz/Mozart etc.

Table 2. Results of computational experiments*

	1	2	3	4	5
Workers × Tasks	5 × 16	5× 24	5 × 28	5 × 32	5 × 36
Number of variables	320	480	560	640	720
Competences changed	12	17	19	21	23
Robust structure determination time [sec.]	1.14	4.18	6.62	10.46	14.75
	6	7	8	9	10
Workers × Tasks	10 × 16	10 × 24	10 × 28	10 × 32	10 × 36
Number of variables	1440	2160	2550	2880	3240
Competences changed	8	11	15	17	19
Robust structure determination time [sec.]	129	436	711	1046	>1000
	11	12	13	14	15
Workers × Tasks	15 × 16	15 × 24	15 × 28	15 × 32	15 × 36
Number of variables	3360	5040	5880	6720	7560
Competences changed	6	5	no data	no data	no data
Robust structure determination time [sec.]	>1000	>1000	>1000	>1000	>1000

*computer parameters: Intel i7-4770, 8 GB RAM

5 Concluding Remarks

The problem of improving workers' competences to make personnel robust to disruptions caused by unexpected personnel absences is rarely discussed in the literature. The declarative model proposed in this study and the constraint programming method used to solve the problem lend themselves well to implementation in commercial DSS software.

One limitation of the model proposed is the assumption that every worker can improve their competences with regard to each task. In practice, it may be that a specific worker is not able (or willing) to acquire the competences needed to realize specific tasks, e.g. because the subject matter of these tasks is out of their scope of interest. Another observation is that different workers might acquire a given competence at a different pace. Then, the problem can be formulated as an optimization problem: which of the alternative variants of the competence structure allow the fastest worker adjustment to guarantee robustness to worker absences? This type of problem should be considered in future research.

Furthermore, it should be noted that the robustness of a competence structure can be obtained by other ways than improving that structure, e.g. by increasing/decreasing employee's hour limits.

In our future work, we plan to focus on robustness of competence structures to other disruptions, such as the loss of employee qualifications (competences), changes in number of tasks, simultaneous (and/or consequent) absence of several employees.

References

1. Antosz, K.: Maintenance–identification and analysis of the competency gap. Eksploatacja i Niezawodnosc–Maintenance and Reliability **20**(3), 484–494 (2018)
2. Banaszak, Z., Bocewicz, G.: Declarative modeling for production order portfolio scheduling. Found. Manage. **6**(3), 7–24 (2014)
3. Bocewicz, G., Banaszak, Z.: Decision Support Driven Models and Algorithms of Artificial Intelligence. Warsaw University of Technology, Warsaw (2011)
4. Coolahan, J.: Key Competencies: A Developing Concept in General Compulsory Education. European Commission, Eurydice, Brussels (2002)
5. Domenech, B., Lusa, A.: A MILP model for the teacher assignment problem considering teacher's preferenes. Eur. J. Oper. Res. **249**(3), 1153–1160 (2016)
6. Gunawan, A., Ng, K.M., Poh, K.L.: A mathematical programming model for a timetabling problem. In: Proceedings of the 2008 International Conference on Scientific Computing, pp. 42–47 (2006)
7. Güler, M.G., Keskin, M.E., Döyen, A., Akyer, H.: On teaching assistant-task assignment problem: a case study. Comput. Ind. Eng. **79**, 18–26 (2015)
8. Lai, C.-M., Yeh, W.-C., Huang, Y.-C.: Entropic simplified swarm optimization for the task assignment problem. Appl. Soft Comput. **58**, 115–127 (2017)
9. Levy-Leboyer, C.: La gestion des compétences (Competence management). Les Editions d'Organisation, Paris (1996)
10. Moreira, J.J., Reis, L.P.: Multi-agent system for teaching service distribution with coalition formation. In: Rocha, Á., Correia, A.M., Wilson, T., Stroetmann, K.A. (eds.) Advances in Information Systems and Technologies. AISC, vol. 206, pp. 599–609. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36981-0_55
11. Patalas-Maliszewska, J., Kłos, S.: An intelligent system for core-competence identification for industry 4.0 based on research results from German and Polish manufacturing companies. In: Burduk, A., Mazurkiewicz, D. (eds.) ISPEM 2017. AISC, vol. 637, pp. 131–139. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-64465-3_13
12. Schutle, H., Smolka, G., Wurtz, J.: Finite Domain Constraint Programming in Oz. German Research Center for Artificial Intelligence, Saarbrucken (1998)
13. Serpell, A., Ferrada, X.: A competency-based model for constriction supervisor in developing countries. Pers. Rev. **36**(4), 585–602 (2007)
14. Spliet, R., Dekker, R.: The driver assignment vehicle routing problem. Networks **68**(3), 212–223 (2016)
15. Szwarc, E.: Ocena odporności planu obsady zajęć dydaktycznych na zakłócenie zmiany liczby grup zajęciowych (in Polish). Res. Enterp. Mod. Econ. Theor. Pract. **3/2018**(25), 135–145 (2018)
16. Szwarc, E., Bach-Dąbrowska, I., Bocewicz, G.: Competence management in teacher assignment planning. In: Damaševičius, R., Vasiljevienė, G. (eds.) ICIST 2018. CCIS, vol. 920, pp. 449–460. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99972-2_37
17. Wikarek, J.: Lecturers' competences configuration model for the timetabling problem. In: Ganzha, M., Maciaszek, L., Paprzycki, M. (eds.) Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, vol. 15, pp. 441–444 (2018)
18. Wikarek, J., Sitek, P.: A data-driven approach to modeling and solving academic teachers' competences configuration problem. In: Nguyen, N.T., Gaol, F.L., Hong, T.-P., Trawiński, B. (eds.) ACIIDS 2019. LNCS (LNAI), vol. 11431, pp. 408–419. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14799-0_35
19. Woodruffe, C.: What is meant by competency? In: Designing and Achieving Competency. McGraw-Hill, New York (1992)