Decision Support Model for the Configuration of Multidimensional Resources in Multi-project Management

Jarosław Wikarek[1], Paweł Sitek[1](✉), and Zbigniew Banaszak[2]

[1] Kielce University of Technology, Kielce, Poland
{j.wikarek,sitek}@tu.kielce.pl
[2] Faculty of Electronics and Computer Science,
Koszalin University of Technology, Koszalin, Poland
banaszak@ie.tu.koszalin.pl

Abstract. In today's competitive knowledge-based economy, the introduction of new solutions, i.e. new products and services, new technologies, new organizational structures, etc., most often requires a project approach. Due to constrained resources, tight deadlines and, usually, a large number of implemented projects, the multi-project environment is used in practice. The key element in multi-project management is appropriate configuration and the use of constrained resources (e.g. machines, tools, software, employees, etc.). Modern resources are characterized not only by their availability and abundance, but also have many additional features that may affect the functionality and configurability of a given resource. Hence, before commencing the implementation of a project, and even more so for a set of projects, managers must answer a few key questions related to such resources, such as: *Do we have resources with proper features/functions to implement the set of projects on the given date and schedule? If not, what resources and features are missing?* etc. Obtaining answers to these types of questions may decide about the success of projects. The paper presents a decision support model for the configuration of multidimensional resources in a multi-project environment, which can be used in both a proactive and reactive approach. Many computational experiments were also carried out to verify the model itself and the methods of its implementation.

Keywords: Multi-project environment · Decision support · Resource allocation · Mathematical programming · Proactive and reactive approach

1 Introduction

The continuous development of companies is related to the implementation of new technological solutions in the IT area, modern organizational forms, new methods of communication and marketing, introducing new products and services to the market, etc. All this means that companies are forced to implement an increasing number of projects at the same time. This trend has resulted in a significant increase in the role of

N. T. Nguyen et al. (Eds.): ICCCI 2021, LNAI 12876, pp. 290–303, 2021.
https://doi.org/10.1007/978-3-030-88081-1_22

multi-project management [1, 2]. The effective implementation of simultaneous projects, very often from different areas of operation, enables the company to quickly develop [3]. Today, most companies manage multiple projects, although they are still looking for the best decision support tools in these complex environments.

Most modern IT tools supporting multi-project management [4] include scheduling, task prioritization, risk management [5], etc. Some also enable resource allocation [6]. Wherein the allocation of resources, if already included in such tools, is usually in the quantitative aspect. Contemporary resources can fulfill various functions. They can also be configurable, extended with new functionalities, etc. This completely changes the approach to the problem of resource allocation. In this case we can talk about multi-dimensional resources, each dimension of which is a specific function/features that the resource carries out. A good example of such a resource is a modern employee, and functions are his competencies. Another example is a computer/microprocessor and the functions are the corresponding software.

In this context it is extremely important to correctly allocate appropriately configured resources to individual project tasks. The problem can be formulated more generally than the allocation itself. *Do we have a sufficient amount of properly configured resources to implement a set of projects? Are any resource features missing*?

The main contribution of the paper is a resource configuration model that takes into account individual features/functions of resources and its implementation using the universal modeling language AMPL (A Mathematical Programming Language) [7]. The model enables decision support in multi-project management in the context of resource allocation and configuration. In particular, decisions regarding project feasibility in the context of available resources, defining missing resources or the need to reconfigure existing resources, etc.

2 Problem Statement and Illustrative Example

The problem of the configuration and allocation of multidimensional resources in multi-project management is discussed. The most basic definition of multi-project management (MPM) is managing an environment in which people are working on multiple projects simultaneously [8]. In such an environment, management has to deal with parallel projects of different dates and sizes, but with a common pool of resources. The very concept of multi-project management is quite new, so there is a lack of strategies, techniques and tools to deal with it effectively. Many project managers complain about a shared pool of resources, which often results in shuffling people and other resources that are used in their projects [9].

The problem of the configuration and allocation of resources in the multi-project environment is defined as follows:

- A set of projects to be performed is given (o – project index $o \in O$). The project consists of a set of tasks (k – task index $k \in K$). Failure to complete any task is tantamount to failure to complete the entire project.

- Each project is characterized by a specific/given schedule of performing tasks, which is known and unchanging. Note: So, the problem under consideration is not a scheduling problem, but a variation and development of the resource allocation problem. This has some practical justification, as in many cases the contracting authority imposes a project implementation schedule.
- Resources with certain features/functions (e – feature index $e \in E$) are needed for each task. The general idea of multidimensional resources is shown in Fig. 1. Table 1 shows sample resources and their features with values in parentheses.
- If several resource features/functions are required to perform a task, a resource with all the necessary features is allocated.
- Generally, resources can be configured, i.e. supplemented with new features (note: not every resource can be supplemented with every feature).
- A feature of a resource can be equated with its dimension. A resource can generally have many dimensions, or it can be supplemented with further dimensions. Therefore, this type of resource is called multidimensional resources.
- Resources may be temporarily unavailable.

Table 1. Resources and their features

Resource	Feature (values)
Employee	Employee competencies (programmer, analyst, welder, miller, etc.)
Computer	Software (SQL database, C #, Java, Phyton, Android, etc.)
CNC	Tools (drill bits, cutters, etc.)

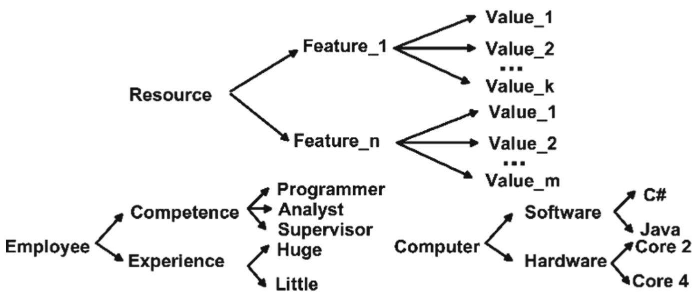

Fig. 1. Model of multidimensional resources/resource-feature-value of the feature

For such a problem, a few basic questions can be formulated (Table 2) which are crucial for the project management process in the context of resources.

Obtaining answers to the above questions is very important in multi-project management and may determine the success of a given set of projects. It also enables efficient management of resources and their rational use. It also facilitates the reaction if there is a lack of a resource or if the resource does not have the needed feature. The proposed questions concern both reactive decisions (Q1A, Q1B, Q2, Q3A, Q4A) and proactive decisions (Q1C, Q3B, Q4B). In order to present the defined problem of the configuration of multidimensional resources let us consider the illustrative example exam_01.

Table 2. Questions regarding resources in multi-project management

Question	Description
Q1A	Is your set of multidimensional resources sufficient to ensure that your set of projects is carried out on schedule?
Q1B	If not, what resources and how many features are missing?
Q1C	How and which resources should be modified and/or new ones acquired?
Q2	If a new project with a given structure and schedule is added to the set of implemented projects, will the available resources be sufficient to implement it without changes to the already implemented projects?
Q3A	Will the lack of a specific $i \in I$ resource enable the implementation of a set of projects?
Q3B	How should the configuration of resources be changed in order to complete the set of projects on schedule and on time when the resource $i \in I$ is unavailable?
Q4A	Will the lack of any $i \in I$ resource enable the implementation of a set of projects?
Q4B	How should the configuration of resources be changed in order to complete the set of projects on schedule and on time in the absence of any resource $i \in I$?

Illustrative Example exam_01. A set of five projects $o_1..o_5$ is given, which should be implemented in accordance with the schedule shown in Fig. 2. As part of the implementation of the above set of projects, tasks $k_1..k_{25}$ should be implemented. Appropriate features $e_1..e_{10}$ of resources are necessary for the implementation of each task. The list of the required features of the resources necessary for the implementation of individual tasks in the orders is presented in Table 3. Table 4 presents the list of resources and their features. A three-value coding was used for this: 1-means that a given resource has a given feature, 0-means that a given resource does not have a given feature but can acquire it, −1-means that a given resource does not have a given feature and cannot acquire it.

For illustrative example *exam_01*, questions Q1A and Q1B were formulated with the data instances as in the tables (Table 3, Table 4).

The answer to question Q1A required a lot of work by the operator who, by manually analyzing the data from Tables 2 and 3, the Gantt chart (Fig. 2), after many attempts stated that the resources with specific characteristics (Table 4) do not allow the implementation of projects $o_1..o_2$ according to the specified schedule (Fig. 2).The answer to question Q1B and the remaining questions, Q2..Q4, despite the small size of *exam_01*, was unsuccessful. It was not possible to answer this question by performing manual analysis with an acceptable amount of work and operator time, e.g. within an hour.

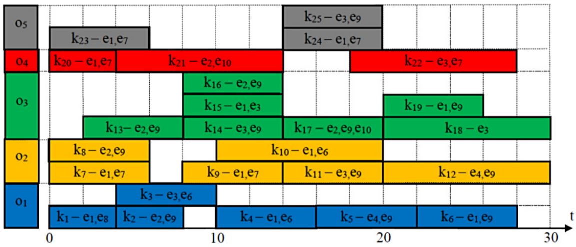

Fig. 2. Schedules for the implementation of project tasks with the required features

Table 3. Required resource characteristics for the implementation of individual project tasks

Project	Task	Time	Resources									
			e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}
o_1	k_1	4	+	–	–	–	–	–	–	+	–	–
	k_2	4	–	+	–	–	–	–	–	–	+	–
	k_3	6	–	–	+	–	–	+	–	–	–	–
	k_4	6	+	–	–	–	–	+	–	–	–	–
	k_5	6	–	–	–	+	–	–	–	–	+	–
	k_6	6	+	–	–	–	–	–	–	–	+	–
o_2	k_7	6	+	–	–	–	–	–	+	–	–	–
	k_8	6	–	+	–	–	–	–	–	–	+	–
	k_9	6	+	–	–	–	–	–	+	–	–	–
	k_{10}	10	+	–	–	–	–	+	–	–	–	–
	k_{11}	6	–	–	+	–	–	–	–	–	+	–
	k_{12}	10	–	–	–	+	–	–	–	–	+	–
o_3	k_{13}	6	–	+	–	–	–	–	–	–	+	–
	k_{14}	6	–	–	+	–	–	–	–	–	+	–
	k_{15}	6	+	–	+	–	–	–	–	–	–	–
	k_{16}	6	–	+	–	–	–	–	–	–	+	–
	k_{17}	6	–	+	–	–	–	–	–	–	+	+
	k_{18}	10	–	–	+	–	–	–	–	–	–	–
	k_{19}	6	+	–	–	–	–	–	–	–	+	–
o_4	k_{20}	4	+	–	–	–	–	–	+	–	–	–
	k_{21}	10	–	+	–	–	–	–	–	–	–	+
	k_{22}	10	–	–	+	–	–	–	+	–	–	–
o_5	k_{23}	6	+	–	–	–	–	–	+	–	–	–
	k_{24}	6	+	–	–	–	–	–	+	–	–	–
	k_{25}	6	–	–	+	–	–	–	–	–	+	–

+ the task requires a certain feature; – the task does not require a certain feature.

Table 4. Features of individual resources

Resource	Features									
	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}
i_1	1	−1	1	−1	−1	0	0	0	1	0
i_2	−1	1	−1	1	0	0	0	1	0	1
i_3	1	−1	1	0	1	1	1	0	0	0
i_4	1	1	−1	0	1	0	1	1	0	0
i_5	1	−1	1	0	0	0	1	0	0	0
i_6	1	−1	−1	0	0	1	1	0	0	0
i_7	1	−1	−1	0	0	0	1	0	1	0
i_8	1	1	−1	0	1	0	0	1	0	0
i_9	−1	0	1	1	0	0	0	0	1	0
i_{10}	−1	1	−1	1	0	0	0	0	0	1

The analysis of this and other similar illustrative examples showed that manual or semi-automatic/e.g. using tools such as spreadsheets /, finding answers to questions Q1..Q4 is very difficult, if not impossible. These findings became the motivation to undertake research on the development of a model to support decisions in the field of the configuration and allocation of resources with specific characteristics to tasks. An additional area of this research involved ways of implementing the model that would enable quick answers to the questions posed.

3 Decision Support Model for the Configuration of Multidimensional Resources

As a result of the problem analysis and the results of illustrative examples, work was undertaken to work out the decision support model for the configuration of multimodal dimensions. Figure 3 shows the general concept of building the model. The model is based on two basic sets, a set of constraints (Table 5) and a set of questions (Table 2). Model parameters are data on resources, their features and schedules. The model was formulated as a constraint satisfaction problem (CSP) [10]. Depending on the question included in the model, the model may take the form of binary integer programming (BIP). The description of the constraints (1).. (14) of the model is presented in Table 5, while the description of decision variables, indexes and parameters is presented in Table 6.

$$\sum_{i \in I} X_{u,i,k} + Y_{u,k} = 1 \ \forall \ k \in K, u \in U \quad (1)$$

$$X_{u,i,k} \leq g_{i,k} \ \forall \ i \in I, k \in K, u \in U \quad (2)$$

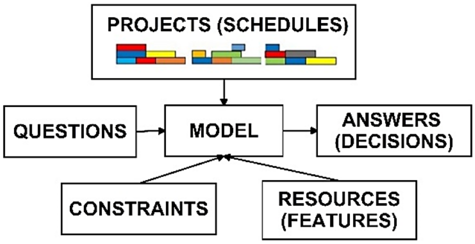

Fig. 3. Model building concept

Table 5. Description and meaning of model constraints

Constraint	Description
(1)	Every task must be done
(2)	If the resource carries out a task, it must have the necessary features to perform it
(3)	In a given state of resource unavailability, tasks are performed only with the use of available resources
(4)	Defines how many unavailability states will fail to complete tasks
(5)	The number of states in which tasks are not completed does not exceed the set value
(6)	Determination of the coefficient $es_{i,k,e}$
(7,8)	Determination of the coefficient $g_{i,k}$
(9)	The resource cannot perform two tasks at the same time
(10)	Resource can be obtained only by permitted features
(11)	The cost of acquiring by resources the features necessary to perform the tasks (Cost1)
(12)	Determining which orders will not be executed
(13)	Penalty for failure to complete orders (Cost2)
(14)	Binary and integrity

$$X_{u,i,k} \leq f_{u,i} \ \forall\, i \in I, k \in K, u \in U \tag{3}$$

$$\sum_{k \in K} Y_{u,k} \leq St \cdot Wx_u \forall\, u \in U \tag{4}$$

$$\sum_{u \in U} Wx_u \leq L \tag{5}$$

$$es_{i,k,e} = (ew_{i,e} + Ex_{i,e}) \cdot ej_{k,e} \forall i \in I, k \in K, e \in E \tag{6}$$

$$(1 - g_{i,k}) \leq \sum_{e \in E} ej_{k,e} - \sum_{e \in E} es_{i,k,e} \forall\, i \in I, k \in K \tag{7}$$

Table 6. Decision variables, indexes and parameters

Symbol	Description
Indexes	
I	Set of resources
O	Set of orders
K	Set of tasks
U	Set of resource unavailability
E	Set of features
i	Resource index ($i \in I$)
o	Order index ($o \in O$)
k	Task index ($k \in K$)
u	The resource unavailability status index ($u \in U$)
e	Resource feature index $e \in E$
Parameters	
$h_{k1,k2}$	If tasks k_1 and k_2 are performed at the same time $h_{k1,k2}= 1$ otherwise $h_{k1,k2}= 0$
$ew_{i,e}$	If resource i has the feature e $ew_{i,e}= 1$ otherwise $ew_{i,e}= 0$
$ep_{i,e}$	If the resource i can get the resource feature e $ep_{i,e}= 1$ otherwise $ep_{i,e} = 0$
$b_{o,k}$	If task k is part of order o $b_{o,k}= 1$ otherwise $b_{o,k}= 0$
r_o	Penalty for failure to complete the order o
$ek_{i,e}$	The cost of obtaining the resource feature e for the resource i
$ej_{k,e}$	If resource feature e is required to perform task k than $ej_{k,e}= 1$ otherwise $ej_{k,e}= 0$
$f_{u,i}$	If resource i is available for unavailable state u than $f_{u,i}= 1$ otherwise $f_{u,i}= 0$
St	Very large constant
Decision variables	
$X_{u,i,k}$	If the resource i performs the task k during the unavailable state ($u \in U$) than $X_{u,i,k}= 1$ otherwise $X_{u,i,k}= 0$
Wx_u	If all tasks k are completed during the unavailable state ($u \in U$) $Wx_u= 1$ otherwise $Wx_u= 0$
$Y_{u,k}$	If task k is not executed during unavailable state ($u \in U$) $Y_{u,k}= 1$ otherwise $Y_{u,k}= 0$
Z_o	If the order o has not been executed $Z_0= 1$ otherwise $Z_0= 0$
$Ex_{i,e}$	If the resource i is required to obtain the feature e to perform the tasks $Ex_{i,e}= 1$, otherwise $Ex_{i,e}= 0$
Determined value	
$g_{i,k}$	If the resource i has the required functionality to perform the task k $g_{i,k}= 1$ otherwise $g_{i,k}= 0$

(*continued*)

Table 6. (*continued*)

Symbol	Description
$es_{i,k,e}$	If the resource *i* has or has acquired the feature *e* needed to complete the task *k* $es_{i,k,e} = 1$ otherwise $es_{i,k,e} = 0$
Cost1	The cost of acquiring by resources the features necessary to perform the tasks
Cost2	Penalty for failure to complete orders
Control parameter	
L	In how many cases of unavailability states *u*, tasks will not be completed (0: in all; 1: in one state of absenteeism, tasks will not be completed; 2: in two states of absenteeism, tasks will not be completed, etc.)

$$ST \cdot (1 - g_{i,k}) \leq \sum_{e \in E} ej_{k,e} - \sum_{e \in E} es_{i,k,e} \forall\, i \in I, k \in K \quad (8)$$

$$X_{u,i,k1} + X_{i,i,k2} \leq 1 \forall i \in I, k1 \in K, k2 \in K \wedge h_{k1,k2} = 1 \quad (9)$$

$$Ex_{i,e} \leq ep_{i,e} \forall i \in I, e \in E \quad (10)$$

$$Cost1 = \sum_{i \in i} \sum_{e \in E} (ek_{i,e} \cdot Ex_{i,e}) \quad (12)$$

$$\sum_{u \in U} \sum_{k \in K} (Y_{u,k} \cdot b_{o,k}) \leq St \cdot Z_o \forall o \in O \quad (12)$$

$$Cost2 = \sum_{o \in O} (Z_o \cdot r_o) \quad (13)$$

$$Z_o, Wx_u, Y_{u,k}, Ex_{i,e}, X_{u,i,k} \in \{0, 1\} \forall\, i \in I, k \in K, u \in U, e \in E, o \in O \quad (14)$$

A significant aspect of the analyzed problem of resource configuration is the fact that the *i* ($i \in I$) resources may be unavailable due to various causes (e.g. breakdown, scheduled inspection, etc.). Assume that, u_j denotes the case of unavailability of selected resources (e.g. $u_1 = \{i_3, i_7\}$ means that we are analyzing a case where the i_3, i_7 resources are unavailable, whereas if $u_2 = \{i_3\}$ it denotes a case where the i_3 resource is unavailable, where $i_3, i_7 \in I$, etc.). U denotes a set of all such analyzed (interesting to us) cases of employee unavailability $u_j \in U$. It is easy to calculate that if we have 5 resources and if we examine all possible single unavailability of resources, the U set will comprise 5 elements. On the other hand, if we want to account for all possible unavailability of two resources, the *U* set will comprise 10 or 15 elements if we also take single unavailability into account, etc. In operating practice, a situation where the *U* set has to be taken into account with elements specifying unavailability of random two, three, four, etc. resources happens very rarely.

4 Computational Experiments

The model was implemented using the universal modeling language AMPL (Appendix A) language and the mathematical programming/constraint programming environment GUROBI [11]. Subsequently, numerous computational experiments were carried out, divided into two stages. In the first stage, all the questions for the illustrative example *exam_01* were answered. The answer to question Q1A was NO, the answer to questions Q1B and Q1C – in order to realize the set of projects, three resources, i_2, i_4, i_8, should acquire feature e_9, and resource i_8 additionally feature e_{10}. Figure 4 shows the appropriate allocation of resources to tasks. In this allocation, resources i_2, i_4, i_8 have new features as per the answer to question Q1C. The next questions are about resources with altered characteristics according to Q1C.

The answer to question Q2 is related to the emergence of new o_6 and o_7 designs. Assume all data are as in illustrative example *exam_01* (including modified resources i_2, i_4, i_8). There will be new projects with a task implementation schedule as in Fig. 5 with the requirements as to the characteristics of resources as in Table 7. The answer to question Q2 is as follows. In order to implement a set of projects together with new projects without changing the already planned assignments of resources to tasks (Fig. 4), resource i_1 should acquire the e_8 feature, resource i_3 should acquire the e_9 feature, and resource i_9 should acquire the e_5 feature. The new method of allocating resources to tasks consistent with the answer to question Q2 is presented in Fig. 6.

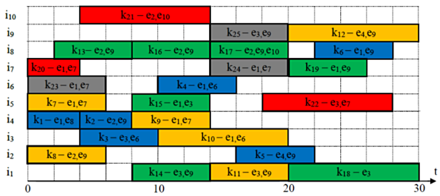

Fig. 4. Allocation of resources for the implementation of tasks

Fig. 5. Schedules for the implementation of additional projects with the required features for the tasks

Question Q3A was formulated as follows: Will the lack of the i_5 resource from t = 8 enable the execution of the order set $o_1..o_5$? The answer to this question is unthinkable. Therefore, question Q3B was asked in the form: *How should the configuration and resource allocation be changed*? Answer: The method of assigning resources to tasks should be changed as shown in Fig. 7.

Table 7. Resource features needed to carry out specific tasks of additional projects

Project	Task	Time	Features									
			e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}
o_6	k_{26}	4	+	–	–	–	–	–	–	+	–	–
	k_{27}	4	–	–	+	–	–	–	–	–	+	–
	k_{28}	4	+	–	–	–	–	–	–	–		–
o_7	k_{29}	4	+	–	–	–	–	–	–	–	–	–
	k_{30}	4	–	–	–	–	+	–	–	–	+	–

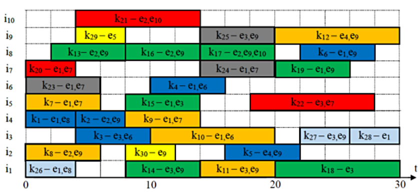

Fig. 6. Allocation of resources for the implementation of tasks, taking into account new projects

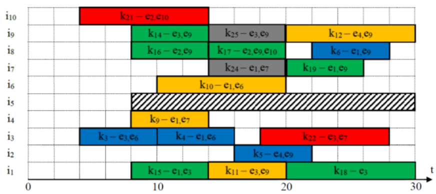

Fig. 7. Allocation of resources for the implementation of tasks (resource unavailability i5 from time t = 8)

The last question, Q4, seems to be the most difficult as it is proactive and requires a lot of checks. They were formulated in the form*: How should resources be configured to be able to implement a set of $o_1..o_5$ projects (according to the schedule in Fig. 2 for resources with characteristics from Table 4) in the event of the unavailability of any resource*? In practice, depending on the resource, the unavailability may result from breakdowns, renovation and inspection, absence, etc. For question Q4 formulated in this way, the obtained answer defines an additional configuration of resources in the form: resource i_3 should acquire feature e_9, resource i_4 should acquire feature e_4, resource i_5 should acquire the e_6 feature and resource i_9 should acquire the e_2 feature. Figure 8 shows two exemplary (there are ten of them all) allocations of resources to tasks in the event of unavailability i_1 and i_4.

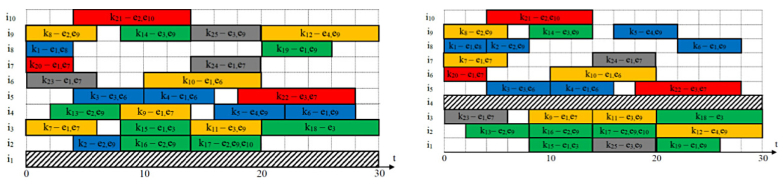

Fig. 8. The allocation of resources to tasks using the resource unavailability i_1 or i_4

The second stage of the computational experiments was to assess the effectiveness of the model implementation method. The experiments were carried out for ten different instances of In1..In10 data (instance In3 is illustrative example examp_01). Individual data instances differed in the number of projects (*N_o*), the number of tasks (*N_k*), the number of resources (*N_z*) and the number of features (*N_e*). This translates into the size of the solved examples, i.e. the number of decision variables (*V*) and constraints (*C*), and, consequently, the computation time (*T*). Questions Q1A (the question required the least effort, simple general question) and question Q4 (the question required the greatest effort of calculations, due to the need to check the unavailability of each of the $i_1..i_{10}$ resources) were selected for the experiments of this phase. The computation times for all data instances for both questions are presented in Table 8.

It turned out that the calculations lasted from 4 to 471 s, which was an acceptable timeframe and confirmed the correctness of the implementation method (AMPL and GUROBI). For data instances of much larger sizes, it is planned to use the proprietary hybrid approach [12] and selected metaheuristics for the implementation of the model.

Table 8. Results of the second phase of computational experiments

In	N_o	N_k	N_z	N_e	Q1A			Q4B		
					V	C	T(s)	V	C	T(s)
In1	3	15	10	5	1127(216)	1469	4	2471(1560)	4763	12
In2	4	20	10	5	1452(271)	1574	7	3271(2060)	5663	24
In3	5	25	10	10	3147(376)	3739	9	5381(2610)	9462	39
In4	7	35	10	10	4357(455)	5269	11	7481(3610)	13963	49
In5	9	45	15	10	5317(571)	9545	13	17736(10290)	32115	67
In6	10	50	15	10	9322(951)	10524	15	19656(11415)	34215	94
In7	10	60	15	15	13674(1156)	17359	17	25171(13740)	43993	189
In8	10	70	15	15	15177(1345)	20069	19	32521(15990)	47943	194
In9	15	80	20	15	27612(1981)	30344	34	57951(32320)	64232	345
In10	15	90	20	15	31022(2191)	32954	43	65151(36322)	91923	471

N_o – number of projects, N_k – number of tasks, N_z – number of resources, N_e – number of features, V – number of decision variables (non-zero), C – number of constraints, T (s) – calculation time in seconds.

5 Conclusions

The model proposed in the paper can be the basis for the design of decision support systems in the scope of the allocation and configuration of resources in multi-project management. Taking into account many features of resources (so-called multidimensions resources) in the model significantly increases the possibility of their configuration and increases the flexibility of assignment to tasks. On the other hand, the proposed model assumes the invariability of project schedules, which significantly simplifies its computational complexity and eliminates the need to additionally solve the problem of task scheduling. One of the advantages of the model is that it can be used in both proactive and reactive modes. Questions Q3B and Q4B are typically proactive in nature, question Q2 can be both proactive and reactive, while the rest of the questions are reactive in nature. All this translates into a wide range of decision support, especially in the area of resource allocation and configuration. In practice, the decision-maker, using this model, can make decisions such as: Is it possible to accept a set of projects for implementation or not? *How can you configure your resources so that the project set can be implemented? How can you configure your resource set so that in the event of the unavailability of any one, the set of projects can be implemented? etc.* Future works will cover the development of the model in terms of: (a) introducing additional questions, (b) the possibility of assigning several resources to the task, (c) adding logical constraints, (d) introducing the costs of resource configuration, and (e) implement the model using a hybrid approach integrating MP, CLP and GA [12, 13]. It is planned to use the proposed model to support decisions, including in the problems of production scheduling [14], vehicle routing problems [15], project management [16], and configuration of employee competences [17].

Appendix A Implementation of the Decision Model in AMPL

```
set Wr; set Jo; set Or; set Co; set Un;
param pa{Wr}; param pb{Wr}; param pk{Jo}; param r{Or};
param wr{Co}; param wc{Co}; param z{Wr,Jo}; param h{Jo,Jo};
param ew{Wr,Co}; param ep{Wr,Co}; param ek{Wr,Co};
param ej{Jo,Co}; param f{Un,Wr}; param b{Or,Jo}; param St;
param L;
var g{Wr,Jo} >=0; var es{Wr,Jo,Co} >=0;
var X{Un,Wr,Jo} >=0, binary; var Y{Un,Jo} >=0, binary;
var Wx{Un} >=0, binary; var Ex{Wr,Co} >=0; var Z{Or} >=0, binary;
var Cost1; var Cost2;
subject to C1 {u in Un, k in Jo}:sum{i in Wr}X[u,i,k]+Y[u,k]= 1;
subject to C2 {u in Un, i in Wr, k in Jo}: X[u,i,k] <= g[i,k];
subject to C3 {u in Un, i in Wr, k in Jo: f[u,i]=0}:
X[u,i,k]<=g[i,k];
subject to C4 {u in Un}:sum{k in Jo} Y[u,k] <= St*Wx[u];
subject to C5 :sum{u in Un} Wx[u] <= L;
subject to C6 {i in Wr, k in Jo, e in Co}:
es[i,k,e]=(ew[i,e]+Ex[i,e])*ej[k,e];
subject to C7a {i in Wr, k in Jo}:
(1-g[i,k])<=(sum{e in Co} ej[k,e]) - (sum{e in Co} es[i,k,e]);
subject to C7b {i in Wr, k in Jo}:
St*(1-g[i,k])>=(sum{e in Co} ej[k,e]) - (sum{e in Co} es[i,k,e]);
subject to C8 {u in Un,i in Wr, k1 in Jo, k2 in Jo:h[k1,k2]=1}:
X[u,i,k1]+X[u,i,k2]<=1;
subject to C9 {i in Wr, e in Co}: Ex[i,e] <= ep[i,e];
subject to C10: Cost1 = sum{i in Wr, e in Co} Ex[i,e]*ek[i,e];
subject to C11 {o in Or}:
sum{u in Un, k in Jo} Y[u,k]*b[o,k] <=St*Z[o];
subject to C15 :Cost2 = sum{o in Or} Z[o]*r[o];
```

References

1. Schwindt, C., Zimmermann, J. (eds.): Handbook on Project Management and Scheduling Vol. 2. IHIS, Springer, Cham (2015). https://doi.org/10.1007/978-3-319-05915-0
2. Walter, M.: Multi-Project Management with a Multi-Skilled Workforce. Springer, Fachmedien, Wiesbaden (2015). https://doi.org/10.1007/978-3-658-08036-5
3. Tonchia, S.: Industrial Project Management, International Standards and Best Practices for Engineering and Construction Contracting, Springer-Verlag GmbH Germany, part of Springer Nature (2018) https://doi.org/10.1007/978-3-662-56328-1
4. Best Project Management Tools & Software for 2021: https://www.proofhub.com/articles/top-project-management-tools-list. Accessed 5 May 2021
5. Sutton, I.: Process Risk and Reliability, Management Operational Integrity Management. Copyright © 2015 Elsevier Inc. https://doi.org/10.1016/C2014-0-01362-7
6. Kane, H., Tissier, A.: A resources allocation model for multi-project management. In: 9th International Conference on Modeling, Optimization & SIMulation, June 2012, Bordeaux, France. hal-00728599f
7. Home-AMPL: https://ampl.com/. Accessed 5 May 2021
8. Kerzner, H.R.: Project Management. John Wiley & Sons. ISBN: 9781119165354 (2017)
9. Maenhout, B., Vanhoucke, M.: A resource type analysis of the integrated project scheduling and personnel staffing problem. Ann. Oper. Res. **252**(2), 407–433 (2015). https://doi.org/10.1007/s10479-015-2033-z
10. Apt, K.: Constraint Logic Programming Using ECLiPSe. Cambridge University Press (2009). https://doi.org/10.1017/CBO9780511607400
11. Gurobi: http://www.gurobi.com/. Accessed 5 May 2021
12. Sitek, P., Wikarek, J.: A multi-level approach to ubiquitous modeling and solving constraints in combinatorial optimization problems in production and distribution. Appl. Intell. **48**(5), 1344–1367 (2017). https://doi.org/10.1007/s10489-017-1107-9
13. Sitek, P., Wikarek, J., Rutczyńska-Wdowiak, K., Bocewicz, G., Banaszak, Z.: Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. Neurocomputing **423**, 670–678 (2021). https://doi.org/10.1016/j.neucom.2020.02.126
14. Nielsen, I., Dang, Q.-V., Nielsen, P., Pawlewski, P.: Scheduling of mobile robots with preemptive tasks. Adv. Intell. Syst. Comput. **290**, 19–27 (2014)
15. Schermer, D., Moeini, M., Wendt, O.: Algorithms for Solving the Vehicle Routing Problem with Drones. In: Nguyen, N.T., Hoang, D.H., Hong, T.-P., Pham, H., Trawiński, B. (eds.) ACIIDS 2018. LNCS (LNAI), vol. 10751, pp. 352–361. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75417-8_33
16. Relich, M.: Portfolio selection of new product projects: a product reliability perspective. Eksploatacja i Niezawodnosc – Maintenan. Reliabil. **18**(4), 613–620 (2016). https://doi.org/10.17531/ein.2016.4.17
17. Szwarc, E., Bocewicz, G., Bach-Dąbrowska, I., Banaszak, Z.: Declarative Model of Competences Assessment Robust to Personnel Absence. In: Damaševičius, R., Vasiljevienė, G. (eds.) ICIST 2019. CCIS, vol. 1078, pp. 12–23. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30275-7_2