Searching For Resilient Project Structure

Judyta Ciemcioch

AGH University of Science and Technology
Cracow, Poland
judyta.ciemcioch@gmail.com

Grzegorz Ginda

AGH University of Science and Technology
Cracow, Poland
gginda@zarz.agh.edu.pl

***Abstract*—Project optimization often deals with simple minimization of its makespan and cost for a given order of project components i.e. project structure. However, project execution effects may be improved at most with the application of appropriate project order. The problem of the utilization of appropriate project structure is nevertheless often neglected. This is why project optimization efforts usually result in suboptimal project implementation. Moreover, the actual effects of optimized project depend on possible disruptions in surrounding environment. The meaningful disruptions may have different e.g. financial and other resource-based, societal, environmenal nature etc. It is necessary, therefore, to make project disruption-proof. This is why a framework for the framework that is capable of delivering project structure that makes project implementation resilient to possible disruptions is presented in the paper.**

Keywords—project management, project structure, disruption, resiliency

I. Introduction

Projects are used in diverse areas to obtain different goals. Successful project implementation depends on careful project preparation and project management. However, the uncontrolled influence of continuously changing surrounding environment, contemporary complex projects are implemented in, may disrupt actual project implementation effects as well. For example, such influence may result from changing fiscal, political, societal and environmental issues. This is why the implementation of contemporary complex projects should be prepared in a way that makes such project resilient to possible disruptions in as much as it is only possible.

Projects are optimized to provide necessary means for the best possible project implementation results. Project optimization is aimed at obtaining the best possible levels of project characteristics – makespan, cost etc. Limits of available resources are included in this regard. The optimization results in a project implementation timeline.

Resulting project timeline deals with the applied order of project components – activities that make obtaining necessary intermediate project implementation results – intermediate goals – possible. An order of project components is called project structure in the paper. A permissible project structure results from pre-order of project components. The pre-order is defined by obligatory precedence of project components.

Possible scale of project optimization depends mainly on the assumed order of project components. The role of order of project components is nevertheless often neglected. And as a result – project optimization results in a suboptimal project implementation only.

Possible project disruption caused by adverse changes in surrounding environment are usually neglected while optimizing a project due to project analysis complexity. Hence the actual appearance of project disruptions result in optimized project implementation performance which is far away from expected performance. Considering possible project disruptions during project optimization becomes important, therefore, to ensure expected project implementation effects that are at least close to the expected effects in the presence of disruptions. It seems that, because to the fundamental role of applied project structure for final performance of project implementation, the application of appropriate choice of the structure would to make project resilient to possible disruption resulting from changes in surrounding environment.

Rising of diverse natural, societal, political, and technical threads cause that resiliency to uncontrollable changes in surrounding environment becomes more and more interesting topic for scientific research [1]. However, up to our knowledge, despite an urgent need for providing reliable tools for resilient planning of project implementation, no proposal currently addresses coping with improving project resiliency to disruptions changes in surrounding environment by means of proper project structure choice. This is why a framework for project optimization which is capable of delivering a project structure that makes project resilient to possible surrounding environment-induced disruptions. Thus, the rest of the paper is structured as follows. The second section is devoted to tentative assumptions. The elements of actual disruption-aware project optimization framework are presented in the third section. Final conclusions are included in the last section.

II. Tentative Assumptions

A project consists of *n* components. The components are related to one another by an obligatory precedence order. The precedence order decides if each pair of project components may be applied only in sequence or in any way. Note that number of possible admissible project structures may rise a lot with the cardinality of a set of project components.

Once the implementation of project component starts it doesn’t stop until its successful end. The same deals is true in the case of the implementation of project components. Each project component is responsible for some intermediate goals. The goal may be achieved by means of using different possible ways (modes). The application of each mode requires utilization of some resources. The resources deal with manpower, equipment, materials, space, financial means etc. Actual availability of resources is limited and may change in time. However, it is assumed that once a given resource is engaged, it becomes entirely involved in the implementation of a project component til its successful completion. The availability of resources may be limited in both time and space. Note that limited nature of resources needed by project components may cause delays in actual start of project component implementation.

Implementation of project components may undergo disruptions resulting from changes in surrounding economic, societal, technological, political, fiscal environment and natural phenomena. Disruptions may result in diverse adverse effects: delays, cost overruns, unnecessary blocking of resources, a need for etc. Possible modes for project component implementation may differ in actual sensitivity to disruptions.

Several attributes may be applied to assess the effects of project implementation. Both tangible attributes (makespan, cost etc.) and intangible attributes (influence on surrounding environment etc.) may be applied in this regard. Level of project attributes results from attributes levels obtained for individual project components. The attributes can be utilised to assess the effects of the implementation of overall project and its components.

Note that clear recommendation of best project implementation involves making several decisions at once. The decisions deal with the indication of appropriate:

- project structure;
- modes for the implementation of project components which also define required resources;
- project starting date.

III. Project Implementation Modelling

A. Princples

The application of flexible and universal means is welcome to model project implementation in a comfortable way. A notion of a joint directed graph (digraph) is applied, therefore, as a basis for description of both obligatory precedence of project components and a project structure. Note that a digraph may be expressed by a matrix. The actual application of such digraph representation facilitates project optimization process a lot.

The digraph of predecessors Γ^- and digraph of successors Γ^+ are applied to express admissible precedence of project components. The digraphs may be expressed by a corresponding binary n by n binary matrix of (project component) predecessors $\mathbf{\Gamma}^-$ and a corresponding n by n binary matrix of (project component) successors $\mathbf{\Gamma}^+$, respectively. The matrices are strictly related to each other:

$$\mathbf{\Gamma}^- = \mathbf{\Gamma}^+ \qquad (1)$$

This is why one of them is sufficient to describe obligatory order of project components.

Structure of project is defined by a digraph $G(U,V)$ and a corresponding n by n matrix $\mathbf{G}$, where U denotes digraph's vertices and V – digraph's arcs. Note that in the case of all above mentioned digraphs vertices (nodes) represent project components while the arcs – the precedence of project components.

A notion of a network $S(G,\Phi,\Psi)$ is applied to express actual project implementation. The network is based on a digraph G. Symbols Φ, Ψ express sets of attributes describing project components and sets of attributes which define their immediate precedence, respectively.

The effects of project implementation are expressed by a set of meaningful attributes. The set may include casual tangible project attributes: makespan, cost, starting date, due date as well as other original tangible attributes e.g. level of the utilization of available resources. The application of intangible project attributes is also possible. All in all, project implementation attributes result from actual network S. However, they may be also directly influenced by changes in surrounding environment.

To provide necessary means for recommending project implementation and assess project implementation quality, a vector function F is introduced. The function makes final recommendation of project implementation(s) possible. The recommendation is results from a multi-level optimization with the following goal function:

$$\min_{G\in\Gamma}\left\{\min_{\theta_0}\left\{\min_{S} F\left[S\left(G,\Phi,\Psi\right),\omega(\tau)\right]\right\}\right\}, \qquad (2)$$

where: Γ denotes a set of permissible project structures, θ_0 is a starting date for actual project implementation, and $\omega(t)$ expresses the influence of surrounding environment which changes in time t.

The goal function is accompanied by a set of constraints imposed on considered attributes of whole project and its components. Considered optimization problem belongs to the general class of multi-criteria multi-mode resource-constrained project scheduling problems [2]. It is nevertheless a peculiar and unique class instance because it considers intangibles and unknown influence of surrounding environment.

It is visible from goal function (2) that consecutive optimization levels deal with individual decisions. The decisions pertain to the choice of appropriate:

- Project structure.
- Starting date for actual project implementation.
- Actual modes for project components.

Note that higher level decisions impose considerable restraints on permissible lower level decisions.

Goal function is intentionally given in a general form (2). This is because such form allows to make use of both tangible and intangible project attributes while optimizing project implementation. Moreover, the general nature of the goal function (2) doesn't favour use of any particular optimization model class and opens presented framework to the wide family of optimization model classes.

B. The optimization

Several challenges arise while considering the optimization of project implementation. The main problems deal with the need for:

- Considering all permitted project structures and modes for project components.
- The use of both tangible and intangible project attributes.
- The influence of surrounding environment.

The problems impede optimization efforts. Monte Carlo simulations were finally chosen to generate permitted project structures, to select actual modes for project components and to simulate changes in surrounding environment. A notion of Pareto efficiency-based dominance and pair-wise comparisons helped to address a need for use of both tangible and intangible project attributes.

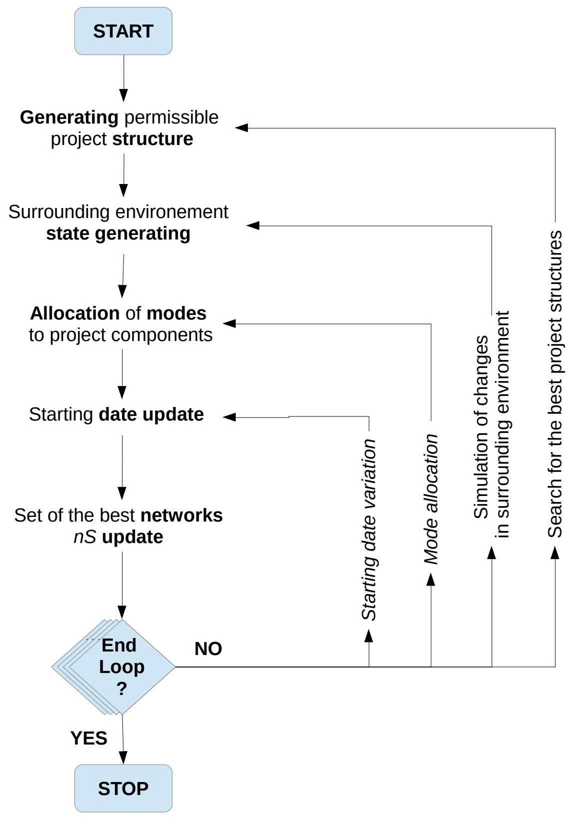

Fig. 1. Illustration of general idea of proposed framework

The general scheme for proposed optimization framework is presented in Fig.1. Four embedded loops are applied in this regard. The outermost loop deals with the generation of permissible project structures. The first inner loop pertains to simulations of changes in surrounding environment. It contains two inner loops. The outer one allows to consider influence of actual allocation of modes to project components on project implementation outcomes. The inner one deals with the influence of starting date θ_0 on the outcomes. Note that replying of calculations for different starting dates makes sense because of possible time-dependent changes in surrounding environment.

The application of each tuple consisting of considered:

- project structure G,
- surrounding environment state $\omega(\tau)$,
- actual mode allocations to project components,
- starting date θ_0.

results in a network *S* corresponding with a distinct project implementation. The pair-wise comparison of the attributes of such project implementation with attributes of previously identified non-dominated project implementation(s) to check if it is a non-dominated in the sense of Pareto-efficiency. If so, it is applied to update the set of non-dominated project implementations. *nS*. Note that the identification of new non-dominated project implementation may also require deepening *nS* update by removing project implementations which become dominated by the newly added project implementation.

Note that to obtain reliable results and facilitate calculations core simulations should be carefully prepared. Several experimental simulation runs are needed, therefore, to identify appropriate probability density models and to indentify necessary parameters e.g. number of required core simulation runs. Particular care is indispensable with regard to the preparation of simulations of surrounding environment state. This is because the reliability of the framework is extremely sensitive to any inadequacy in capturing surrounding environment state reality [3]. Multiple repetitions of optimization framework is also recommended to consider as much possible permitable project implementations as possible.

The optimization of project implementation may result in one or a set of several non-dominated candidates for the recommended project implementation. In the latter case there appears problem which dominated project implementation to recommend for final application? Pair-wise comparisons may prove helpful in this regard, again. The ability to consider difference in importance of project implementation attributes is nevertheless needed here.

For example, a multi-attribute value theory-based technique like Saaty's AHP [4,5] may prove to be a relatively easy to use tool. Another possibility deals with the application of a outranking-based technique like Bran's PROMETHEE [6]. There are also some well known strategies of psychological origin available e.g. intuitionistic heuristic techniques like Tversky's semi-lexoicogrpahic strategy and aspect-wise elimination strategy [7,8]. Note that some additional limitations imposed on main or auxiliary project implementation attributes may also help a lot in the identification of the most valuable project implementation.

To avoid pitfalls with regard to the identification of the really best non-dominated project implementation alternative, the application of sensitivity analysis is recommended. The analysis can deal with the influence of different decision support tools or differences in preferences toward different project implementation attributes.

A general two-stage framework for final recommendation of project implementation (Fig.2) consists of two stages which are devoted to:

- The identification of non-dominated project implementations.
- Final indication of the most advantageous non-dominated project implementation.

C. *Software implementation*

Complexity of proposed framework makes software application support indispensible. Fortunately, the development in information and communication technology delivers a lot of possible options available that are capable of facilitating software implementation of the framework. Many of them are freely available under free and libre open source software (FLOSS) framework. For meaningful details consult for example the *Floss for Science* initiative presented at https://flossforscience.com.

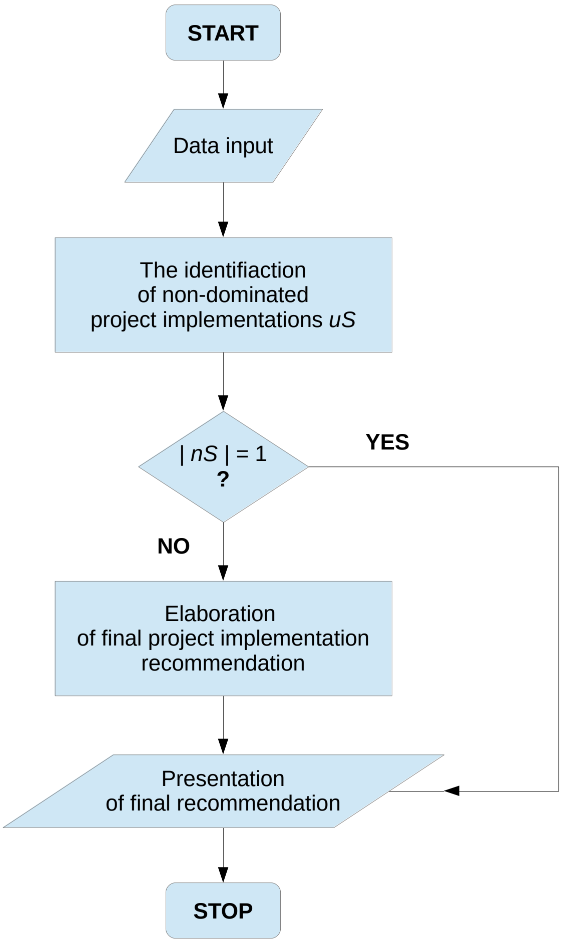

Fig. 2. General scheme of the proposed recommendation framework

There are also several FLOSS options available that seem suitable for the implementation of the proposed framework. For example example, GNU OCTAVE - a multi-platform scientific programming language system available at https://www.gnu.org/software/octave provides necessary means for matrix, numerical and simulation analysis.

Another suitable option is provided by core programming languages with useful extensions. It seems that, due to universality and rising popularity, the application of van Rossum's Python programming language should be recommended in this regard. Python implementations are freely available at web page dedicated to the language: https://www.python.org.

IV. CONCLUSIONS

Contemporary projects are implemented in specific multi-dimensional surrounding environment. The complexity of interactions with surrounding environment result in a considerable dependence of actual project implementation outcomes on actual changes in surrounding environment. Therefore, it is necessary to plan the implementation of contemporary projects in a way which would make them resilient to possible changes in surrounding environment as much as only possible. A framework is thus presented in the paper which is capable of recommending project implementation which would be resilient to changes in surrounding environment at the highest possible level. The framework makes use of appropriate choice of project structure in this regard.

Besides the capability of including tangible and intangible influence of surrounding environment changes, the main merits of the framework cover the ability to include both tangible and intangible effects of project implementation. The framework is also capable of including different possible ways for implementing project components while considering limited availability of necessary resources. Therefore, it seems also to be a tool that considerably improves to the reliability of solutions of wide class of multi-criteria multi--mode resource constrained project scheduling problems.

Universal and comprehensive nature of proposed framework makes it well suited for recommending reliable project structure and a resulting schedule in different areas. Actual reliability of indispensable software implementation of the framework heavily depends, however, on the adequacy of modeling influence of surrounding environment. Specific implementation of the framework requires, therefore, careful adjustment to actual needs. Hopefully, the application of available FLOSS tools makes it approachable.

ACKNOWLEDGMENT

The authors wish to thank AGH UST for providing financial means for the research and publication of the paper.

REFERENCES

[1] Y.-P. Fang, E. Zio, „An adaptive robust framework for the optimization of the resilience of independent infrastructures under

natural hazards", European Journal of Operational Research, vol. 276(3), 2019, pp. 1119–1136.

[2] J. Węglarz, J. Józefowska, M. Mika, G. Waligóra, "Project scheduling with finite or infinite number of activity processing modes a survey", European Journal of Operational Research, vol. 208(3), 2011, pp. 177–205.

[3] M. Dytczak, G. Ginda, N. Szklennik, and T. Wojtkiewicz, "Weather Influence-Aware Robust Construction Project Structure", Procedia Engineering, vol.57, pp. 244–253, 2013.

[4] T.L. Saaty, The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. McGraw-Hill, 1980.

[5] E. Rokou, K. Kirytopoulos, "Multicriteria Decision Making for Project Scheduling under Reource Constraints", Proceedings of ISAHP 2013, June 2013 [paper no.056, available at: https://doi.org/10.13033/isahp.y2013.056] .

[6] J.P. Brans, L'ingénierie de la décision: élaboration d'instruments d'aide à la décision. La méthode PROMETHEE. Presses de l'Université Laval, 1982.

[7] A. Tversky, "Intransitivity of preference", Psychological Review, no. 76, 1969, pp. 31–48.

[8] A. Tversky, "Elimination by aspects: a theory of choice", Psychological Review, no. 79, 1972, pp. 281–299.