Autonomous Team Role Selection On Flexible Projects

Zsolt Kosztyan[1], Peter Harta[2]
[1] Department of Quantitative Methods, Institute of Management, Faculty of Business and Economics, University of Pannonia, Hungary
[2] Department of Quantitative Methods, Institute of Management, Faculty of Business and Economics, University of Pannonia, Hungary

Abstract: In this study, a matrix-based project planning model that can handle flexible projects, employee synergies, and the distinction between hard and soft skills is proposed. Traditional scheduling methods cannot capture the importance of flexible policies such as those found in agile projects. The proposed method includes autonomous team role selection and is evaluated through a real-life case study. The findings show that positive synergies between employees can decrease project makespan, and the best solution can be achieved through autonomous team role selection. This research expands on the synergy-based software project scheduling problem (SPSP) to show how the proposed model can effectively manage team roles and the synergies among employees. The results suggest that the proposed method outperforms the existing nonsynergy-based project scheduling methods, and the use of this approach can help project managers to better understand the importance of such policies and optimize their project schedules accordingly.

Keywords: project scheduling problems; teams; synergy

1 Introduction

In Industry 4.0 (I4.0), the role played by Information Technology (IT) is increasing. I5.0 may mean that software is no longer made by humans but rather by machines. Today's software products, however, must be made by a collaboration of closely cooperating developers (Oza et al., 2013). The importance of software development is also underscored by the fact that a widely studied (Vega-Velázquez et al., 2018) so-called software project scheduling problem (SPSP) deals specifically with the optimal selection of human resources in a software project environment (SPE). This problem is very similar to the multiskilled resource constrained project scheduling problem (MS-RCPSP), and it has similar limitations and shortcomings (Arashpour et al., 2018). In both the case of SPSP and that of MS-RCPSP, the fixed or predefined task duration is dependent on the skill levels of the employees involved (Alba and Chicano, 2007). In addition, the cooperation, or synergies, among employees are neglected. However, the complexity of the various methods used to address these issues strongly limits their practical applicability.

Software applications are now primarily developed flexibly through agile and hybrid approaches and projects (Wysocki, 2019), where the precedence of development processes is not always fixed (Aslam and Ijaz, 2018). Established tasks are prioritized in cooperation with the customer. Task priority determines whether a task is ultimately implemented in this or subsequent development (Al-Saqqa et al., 2020). In addition, agile approaches heavily rely on the continuous communication and collaboration between developers, and the synergistic effect of such communicative collaboration can significantly reduce the makespan of software development projects (SDPs) (Winter, 2015).

The main idea behind SPSP is that human resources are assigned skill abilities that enable them to perform a particular task (Alba and Chicano, 2007). Later researchers (Vega-Velázquez et al., 2018) have further investigated how the level of these skills affects the time necessary to perform the task. Moreover, skills can vary over time, even when they are based on learning and forgetting factors (Vega-Velázquez et al., 2018). In most models, the skill level of agents directly affects the task duration. A higher skill level leads to a shorter task duration, but employee skills cannot be added. Rather than skills, which cannot be summarized, (Kosztyán et al., 2022) suggested the use of work ability or skill performance. In addition, it is much easier to set minimum constraints for skill performance than it is to set those for the skills themselves. This model supports agile approaches to a much greater degree, where one of the requirements is cross-functionality, as based on the agile manifesto; i.e., it is required that resources have similar abilities to replace each other.

Nevertheless, it should be noted that such skill performances, which can eventually be quantified, are mainly meaningful for hard skills. For soft skills, such as communication skills, it is not possible (or only possible with great difficulty) to specify a skill performance. Thus, in our view, both ideas are relevant in modeling. The synergy among employers and its relation to performance have been extensively studied (e.g. Larson, 2009); however, to our knowledge, Kosztyán et al. (2022) is the only study that considers the synergy among employees, and they propose a new class of SPSP that they term the synergy-based software project scheduling problem (SSPSP). Kosztyán et al. (2022) investigated artificial projects and synergy networks to show the effects of synergies. However, this evaluation was not conducted on a real-life example. Putting the theoretical method into practice by identifying the synergistic effects among employees is a tremendous challenge.

We recognize that SPSP has been supplemented through the consideration of different personality types (Stylianou and Andreou, 2012). Numerous studies (e.g. LePine et al., 2011) have shown that synergy is strongly influenced by personality types and team roles. Before the introduction of the agile method and the recognition of the importance of small teams, the Myers–Briggs indicator was more likely to be used. In addition, many articles (e.g. Flores-Parra et al., 2018) reference the Belbin team selection in conjunction with the popular Myers–Briggs indicator (Capretz, 2003) or the Big-five factor traits (Peeters et al., 2006). In contrast, the DISC personal assessment tool is more flexible due to the small number of different personality types (only 4). Because of its simplicity and popularity, the DISC personality types have gradually increased in recent years and are suitable for use in the selection of small agile teams (Lykourentzou et al., 2016).

The DISC personality model was created in 1928 by William Martson, who studied the behavioral differences among individuals. Each letter of the acronym represents a separate personality type along the two dimensions (openness and orientation) on which the method is based. these personality types are Dominance (D), Influence (I), Steadiness (S), and Conscientiousness (C). Each personality type is further matched to a color. D is red, I is yellow, S is green, and C is blue. Although, on a certain level, everyone can exhibit several personality types, each person can be categorized into the personality type that primarily characterizes them. A brief description of the personality types is given in Table 1.

Table 1: Brief description of the DISC personality types

Personality type	Characterization	Main properties
Dominance	extrovert, task-oriented, the leader of the team, takes responsibility	active, powerful, confronting, obstinate
Influence	extrovert, people-oriented, the soul of the team, maintains motivation	friendly, optimistic, easy-going, unstable
Steadiness	introvert, people-oriented, the parent of the team, maintains stability	team-player, reliable, loyal, retractive
Conscientiousness	introvert, task-oriented, the brains of the team, precisely promotes solutions	accurate, analytical, perfectionist, mistrustful

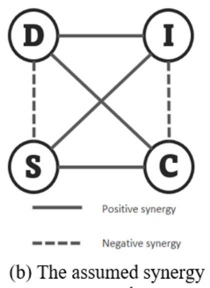
While the optimal allocation of the different DISC personality types can highly influence team performance and well-being (Lykourentzou et al., 2016), another study (Antoniou, 2019) found that there is no relation between balanced and unbalanced teams. Since we distinguish between a synergy network (a measure of joint work) and a sociogram (a measure of social relations), we can conclude that balanced DISC groups exert a strong impact on the synergy network among team members. Based on the literature (Scullard and Baum, 2015), we estimate the positive and negative synergies between DISC personality types in a balanced DISC group, as shown in Figure 1. We can also conclude that the sociometric analysis in a balanced DISC team reveals only positive relations between the team members because members fill the team with heterogeneous personalities. However, in this article, we focus only on the benefit of the project, which is represented by the synergy network.

The synergy network shows positive pairwise synergy between the D and I, D and C, I and S, and S and C personality types, and it shows negative pairwise synergy between D and S and I and C personality types.

1st type	2nd type	Benefit from the 1st type	Benefit from the 2nd type
D	I	leadership and direction	inspire and motivate others
D	C	focus on achieving results	attention to detail
S	I	stability and consistency	build relationships
S	C	work collaboratively	attention to detail

1st type	*2nd type*	Loss from the 1st type	Loss from the 2nd type
D	S	delegate and make pressure	inflexible
C	I	standoffish	open to discuss

(a) Pairs with positive and negative synergy



(b) The assumed synergy network

Figure 1: Synergies of DISC pairs, (a): details of the synergies, (b): the assumed synergy network between DISC personalities

In this study, we use the DISC personality types to determine the synergy network because of the greater simplicity in measuring only four personality types and because of the extensive study of their effects. Our proposed study's contribution to the literature is twofold. (1) The proposed method enables the modelling of both soft and hard skills in flexible projects. (2) The proposed model shows a method for investigating the role of autonomous or directed team-role selection. The proposed SSPSP modification reveals a method for modelling an agile project, where (A) completion priorities can be assigned to tasks; (B) flexible dependencies can be managed; (C) synergies and the team roles of employees can be managed; and (D) soft skills and hard skill performances can be separated. The extended model can manage binary, ordinal, nonadditive (soft) skills, and additive (hard) skill performances. The DISC profiles of employees are used to obtain the expected synergies between developers (see Figure 1). After the specification of team roles, the effect of the team role selection mechanisms can be investigated, and the following research question (RQ) can be answered.

RQ: What type of team role selection mechanism can provide the shortest project duration considering the constraints?

2 Methods

The SSPSP specifies a flexible project plan modeled in a multidomain mapping (MDM) matrix. Kosztyán et al. (2022) proposed a six-domain matrix model. The proposed version of the synergy mapping matrix (SMM) contains six domains and two column vectors. The modified SMM matrix is an $m + n \times m + s + n + 1$ matrix, where the number of employees is *m*, the number of skill performances is s, and the number of tasks is n.

1. The first domain (**Y**), which is an *m* by *m* submatrix, represents the synergy between employees. In the case of $i \neq j$, $[\mathbf{Y}]_{i,j} > 1$ represents positive (or favorable) synergy, $[\mathbf{Y}]_{i,j} = 1$ represents neutral synergy and $0 < [\mathbf{Y}]_{i,j} < 1$ represents negative (or unfavorable) synergy between employees *i* and *j*.
   Furthermore, we assume $[\mathbf{Y}]_{i,i} = 1$ and $[\mathbf{Y}]_{i,j} = [\mathbf{Y}]_{j,i}$.
2. The second domain is the skill domain (**S**). The skill domain is an m by s submatrix, where every skill or skill performance is a nonnegative number ($[\mathbf{S}]_{ij} \in \mathbb{R}_0^+$). $[\mathbf{S}]_j \coloneqq [\mathbf{S}]_{.j} \coloneqq \left[[\mathbf{S}]_{1j}, [\mathbf{S}]_{2j}, \dots . [\mathbf{S}]_{mj}\right]$ represent skill vectors. We suppose that there are h hard skills, as represented by the $[s_1, s_2, \dots, s_h]$ hard skill performance vector, and there are $s - h$ soft skills, as represented by the $[s_{h+1}, \dots, s_s]$ skill vector. $[\mathbf{S}]_{ij} = 0$ means that employee *i* has no work ability for skill *j*. **S** captures either *binary* or *cardinal* levels of skills. In the case of the binary representation, $[\mathbf{S}]_{ij}$ is either 0 or 1. If the levels of skills (e.g., level of language skill, level of communication skill, etc.) are represented, then $[\mathbf{S}]_{ij} \in \mathbb{N}$. These binary and ordinal skills are nonadditive, or in other words, there is no meaning for the sum $\sum_i [\mathbf{S}]_{ij}$. Cardinal skill performances (usually hard skill performances) are additive, and thus $[\mathbf{S}]_{ij} \in \mathbb{R}_0^+$. Synergies can modify skill performance. Let $\boldsymbol{\varepsilon}$ be a subset of employees; then, the *skill* of $\boldsymbol{\varepsilon}$ is:
   $$S_j^{\varepsilon} \coloneqq \bar{Y}_{\varepsilon} * \sum_{i \in \varepsilon} [\boldsymbol{S}]_{ij} \quad (1)$$

   where $\bar{Y}_{\varepsilon}$ is the *geometric mean* of synergies:

   $$\bar{Y}_{\varepsilon} \coloneqq \begin{cases} 1, & if\ |\varepsilon| \leq 1 \\ \sqrt[\eta]{\prod_{i,j \in \varepsilon} \prod_{i<j} [\boldsymbol{Y}]_{i,j}} \quad where\ \eta = \frac{|\varepsilon| \cdot (|\varepsilon| - 1)}{2}, & if\ |\varepsilon| > 1 \end{cases} \quad (2)$$
3. The third domain is a matching domain (**M**). **M** is an *m* by *n* domain, where $[\mathbf{M}]_{ij} \in [0,1]$ represents the maximal relative amount of assignments of employee *i* to task *j*. If $[\mathbf{M}]_{ij} = 0$ ($[\mathbf{M}]_{ij} = 1$), then employee *i* is not (fully) assigned to task *j*.
4. The fourth domain is the activity or task domain (**A**). The activity domain is an *n* by *n* square matrix ($[\mathbf{A}]_{ij} \in [0,1]$), where the diagonal represents the relative priorities of task (activity) completion. $[\mathbf{A}]_{ii} = 1$ represents the mandatory task, and $0 < [\mathbf{A}]_{ii} < 1$ represents the supplementary task. A higher $[\mathbf{A}]_{ii}$ value in the diagonal represents a greater priority (greater score) value. Mandatory tasks cannot be postponed, but lower priority tasks, depending on the constraints, can be postponed to a later project (or a later subproject, which is called a sprint in agile project management). A postponed task's precedence and demands are also neglected. $[\mathbf{A}]_{ij}, i \neq j$ represents the precedence between tasks a_i and a_j (either "a_i must end before a_j starts" ($a_i \prec a_j$), or there is "no precedence between a_i and a_j" ($a_i \sim a_j$)). $[\mathbf{A}]_{ij} = 1$ represents the fixed dependency between task *i* and task *j* ($a_i \prec a_j$ or $a_i \sim a_j$), while $0 < [\mathbf{A}]_{ij} < 1$ represents flexible dependency ("$a_i \rhd\lhd a_j$") between them, which, depending on the constraints, can either be prescribed or relaxed. Importantly, after the optimization and depending on the relevant constraints, every $0 < [\mathbf{A}]_{ii} < 1$ and $0 < [\mathbf{A}]_{ij} < 1$ value must be either 1 or 0, and $[\mathbf{A}]_{ij} = 1$ must imply $[\mathbf{A}]_{ii} = 1$ and$[\mathbf{A}]_{jj} = 1$. This means that the proposed algorithm has to be used to decide which supplementary tasks must be completed or postponed and which flexible tasks must be prescribed or relaxed.
5. The fifth domain is the skilled-word domain (**W**). **W** is the *n* by *s* matrix. $[\mathbf{W}]_{ji}$ stores the required skilled work of skill *i* for task *j*. In the case of binary and ordinal skills, $[\mathbf{W}]_{ji}$ can represent a minimum requirement of skills for completing task *j*, e.g., the minimum level of communication, minimum level of language skills, etc., while in the case of skill performances $[\mathbf{W}]_{ji}$ represents the minimal amount of skilled-work, such as tested functions, documented programming codes, etc.

6. The last domain is the output domain ($\mathbf{O}$), which contains the solution of the SSPSP algorithm. $\mathbf{O}$ is an *n* by *m* matrix (of nonnegative real numbers), where the element $[\mathbf{O}]_{ji} > 0$ represents the (final) allocation of employee *i* to task *j*. The value $[\mathbf{O}]_{ji}$ is the proposed *ratio* of the working time of e_i allocated to a_j; clearly, $[\boldsymbol{O}]_{ji} = 0$ represents *no* allocation. $[\mathbf{O}]_{ji} \leq [\mathbf{M}]_{ij}$ and $\sum_{j:=1}^{n}[\mathbf{O}]_{ji} \leq 1$ must hold for each $j = 1,2,\ldots,n$ and $i = 1,2,\ldots,m$, while $\sum_{j:=1}^{n}[\mathbf{M}]_{ij} \leq 1$ is *not* required for any $i = 1,\ldots,m$.
7. The modified SMM matrix contains two extra column vectors. The first, C, is an *m* by *1* column vector containing the salary of employees, and the second, $\mathbf{T}$, is an *n* by *1* column vector containing the scheduled start time of the tasks.

Figure 2 shows an example of a filled modified SMM matrix, where the number of employees is five and the number of tasks is six (four mandatory tasks and two supplementary tasks). The example represents one binary, one ordinal skill and two skill performances. The symbol "?" represents the variable cells that must be optimized using the proposed algorithm.

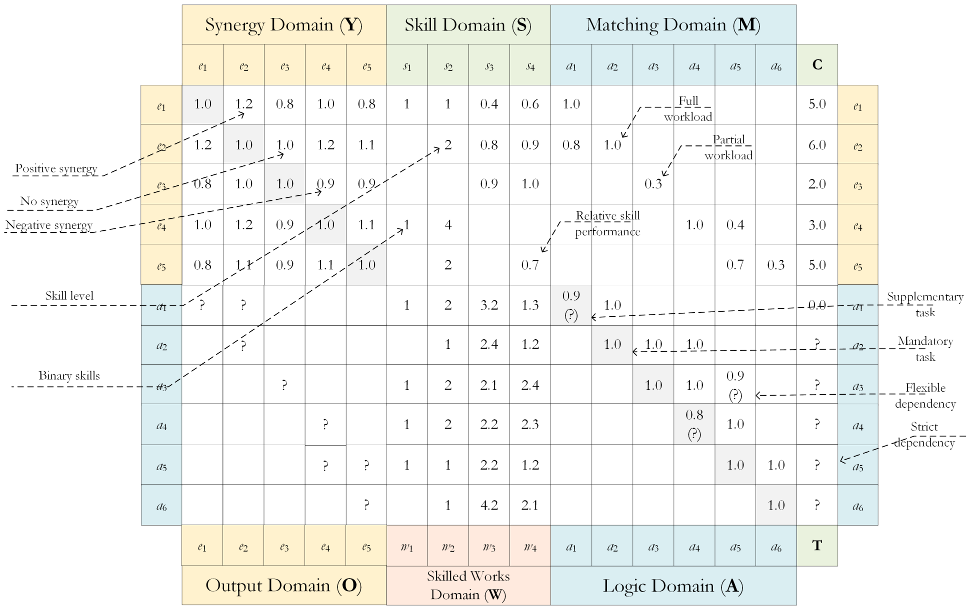


Figure 2: The proposed, modified SMM matrix

The *duration* of activity a_j is denoted by $a_j^{dur}(\mathbf{O})$. The *starting time* of a_j is $a_j^{start}(\mathbf{O})$, and the finishing time is $a_j^{end}(\mathbf{O}) = a_j^{start}(\mathbf{O}) + a_j^{dur}(\mathbf{O})$. The *duration of the project* is denoted by p_{dur} or TPT, and its cost is p_{cost} or TPC. The monthly *salary* of employee e_i is denoted by e_i^{salary} or $[\mathbf{C}]_i$. Since task a_j *requires* $[\mathbf{W}]_{jk}$ skilled work, the *required time* (duration) to fulfill requirement (skill) *k* of task *j* without synergies is:

$$a_{j,k}^{dur}(\boldsymbol{O}) = \frac{[\boldsymbol{W}]_{j,k}}{\sum_{i=1}^{m}([\boldsymbol{S}]_{i,k}*([\boldsymbol{O}]_{j,i})} \tag{3}$$

and the *adjusted required time* (with synergies) is:

$$\boldsymbol{a}_{\boldsymbol{j},\boldsymbol{k}}^{\boldsymbol{dur},\boldsymbol{adj}}(\boldsymbol{O}) = \frac{[\boldsymbol{W}]_{j,k}}{Y_{\varepsilon_j}*\sum_{i=1}^{m}([\boldsymbol{S}]_{i,k}*([\boldsymbol{O}]_{j,i})} \tag{4}$$

where $\varepsilon_j := \{i: 0 < [\mathbf{O}]_{j,i}\}$ is the set of employees finally assigned to task *j*. Without considering the synergies, the duration time of task *j* is:

$$a_j^{dur}(\boldsymbol{O}) = \max_{0<[\boldsymbol{W}]_{j,k}}\{a_{j,k}^{dur}(\boldsymbol{O})\} \text{ and with synergies: } \tilde{a}_j^{dur}(\boldsymbol{O}) \coloneqq a_j^{dur,adj}(\boldsymbol{O}) = \max_{0<[\boldsymbol{W}]_{j,k}}\{a_{j,k}^{dur,adj}(\boldsymbol{O})\} \tag{5}$$

Durations are used to calculate the finish times of the activities $a_j^{end}(\mathbf{O}) = a_j^{start}(\mathbf{O}) + \tilde{a}_j^{dur}(\mathbf{O})$, where:

$$a_j^{start}(\mathbf{O}) \geq \begin{cases} 0, & if \nexists\, a_i \in A,\ a_i \prec a_j \\ max\{a_i^{end}(\mathbf{O}) :\ a_i \prec a_j\} & otherwise \end{cases} \tag{6}$$

The values calculated above enable calculating the duration of the project as follows:

$$TPT_{nosyn} := max\{a_j^{end}(\boldsymbol{O}) : j = 1, \dots, n\} \quad \text{and} \qquad TPT_{syn} := max\{\tilde{a}_j^{end}(\boldsymbol{O}) : j = 1, \dots, n\} \qquad (7)$$

If the decision is made to postpone all supplementary tasks, all flexible tasks are excluded (formally, $[\mathbf{A}]_{min} = [\mathbf{A}]$), and in the case of full assignments (formally, $\mathbf{O} = \mathbf{M}^{\mathrm{T}}$), we obtain a minimum TPT ($\mathrm{TPT_{min}}$). However, in practice, this is usually not feasible, given the constraints. The maximal TPT is infinite when there is no assignment. The cost of the project (TPC) can be calculated as the sum of the salaries of those employees who are paid for their dedication to the project. Since positive synergy reduces (and negative synergy increases) the duration a_j^{dur} to $\tilde{a}_j^{dur}$, the project cost can be calculated with and without the synergy effect, resulting in TPC_{syn} and TPC_{nosyn}, respectively. Formally:

$$TPC_{syn} = \sum_{i=1}^{m} \sum_{j=1}^{n}([\boldsymbol{C}]_i \times [\boldsymbol{O}]_{j,i} \times \tilde{a}_j^{dur}(\boldsymbol{O})) \qquad (8)$$

$$TPC_{\boldsymbol{nosyn}} = \sum_{i=1}^{m} \sum_{j=1}^{n}([\boldsymbol{C}]_i \times [\boldsymbol{O}]_{j,i} \times a_j^{dur}(\boldsymbol{O})) \qquad (9)$$

The maximal amount of costs $\mathrm{TPC_{max}}$ occurs in the case of full assignment and when all supplementary tasks are determined to have been completed. The minimal value is 0 if there is no assignment to any task. The TPS is not influenced by the synergy. Rather, it depends only on the set of decided-to-complete tasks, as denoted by A .

$$TPS := \sum_{i \in A}[\boldsymbol{A}]_{ii} \qquad (10)$$

$\mathrm{TPS_{min}}(\mathrm{TPS_{max}})$ occurs when all supplementary tasks are postponed (completed).

2.1 Target functions

Here, we declare the objective functions that we aim to *simultaneously* optimize by applying the algorithm:

$$TPT \to min;\ TPC \to min;\ TPS \to max \qquad (11)$$

These objective (target) functions can be considered as a multiobjective problem or a composite objective (target) function and can be specified as follows (here, C_s, C_p, C_c and C_t are given reasonable constants):

$$z := 1 - \sqrt[3]{\left(\frac{C_t - TPT}{C_t - TPT_{min}}\right) * \left(\frac{C_c - TPC}{TPC_{max}}\right) * \left(\frac{TPS - C_s}{TPS_{max} - C_s}\right)} \ \to min \qquad (12)$$

we assume the constraints $\boldsymbol{CR_1} - \boldsymbol{CR_6}$ below.

2.2 Constraints

$\boldsymbol{CR_1}$: The employment of each employee e_i in the *project* is not allowed to exceed its *maximum* value:

$e_i^w := \sum_{j=1}^{n}[\mathbf{O}]_{j,i} \leq e_i^{max\,w} := \sum_{j=1}^{n}[\mathbf{M}]_{j,i}$. Clearly, $0 \leq e_i^w \leq 1$ by$\sum_{j=1}^{n}[\mathbf{O}]_{j,i} \leq 1$. (See the matching domain (**M**) in Figure 2) In addition, each activity must be performed by at least one human resource.

$\boldsymbol{CR_2}$: The set of skills that are required to complete an activity must be a subset of the total skills of the employees who actually perform this activity. There are further constraints in SSPSP for managing flexible projects and specifying the set of implemented tasks.

$\boldsymbol{CR_3}$: The TPS must be greater than a specified (score) constraint C_s, or more formally: $TPS < C_s$.

The following three additional constraints are the constraints of the project plan:

$\boldsymbol{CR_4}$: Overwork is allowed up to a *certain level* (roughly: $E^w = \sum_{i=1}^{m} e_i^w \leq K^w$ for some constant K^w, with minor exceptions).

$\boldsymbol{CR_5}$ $(\boldsymbol{CR_6})$: The TPC (TPT) must be less than the cost (time) constraint (C_c) (C_t).

2.3 Applied hybrid genetic algorithm

The SPSP is NP-hard (Xiao et al., 2013), which is a special case of the SSPSP. Kosztyán et al. (2022) showed that the SSPSP is also NP-hard. Kosztyán et al. (2022) also proposed a hybrid genetic algorithm (HGA) for solving the SSPSP problem. The suggested HGA has two phases. In the first phase, a GA is employed to specify the set of completed tasks and the final precendences. In phase 2, a Nelder–Mead method (NMM) is used to refine the schedule start time (SST) to balance the research demands. We extended and parallelized this algorithm, which is also implemented in MATLAB. The original HGA settings are published in Kosztyán et al. (2022); therefore, we focus only on the extensions and modifications. A chromosome encodes a probable solution of SSPSP. The proposed modification of the original chromosome structure enables the selection of a synergy network from a pool. Synergy networks are influenced by team roles. N different team roles specify N different synergy networks. In addition, the proposed method categorizes the skills into binary and ordinal skills, which are usually considered soft skills, and additive skill performances, which are usually used to describe hard skills. Finally, we organized chromosomes into a multichromosome structure (see Figure 3 to decrease the computational time and to share the best chromosomes in parallel computations). The proposed chromosome structure has four parts.

The first chromosome element is the selected number for the synergy network. This number is an ordinal value from 1 to N. The second part is a binary sequence of the decision outcome of completing supplementary tasks and flexible dependencies. The length of this part of the chromosome is $n_F = n_s + n_f$, where n_s is the number of supplementary tasks, and n_f is the number of flexible dependencies. The third part of the chromosome encodes the assignment ratios from the output domain. These real values must be included in the interval. The number of assignment ratios (n_A) is the number of nonzero elements derived from the match domain (**M**). The last part encodes the SST of tasks. The number of elements in this part is n. Therefore, the number of elements of a chromosome vector is $N_c = 1 + n_F + n_A + n$. All four parts of the chromosome have a different type; therefore, different crossover, mutation, and selection mechanisms must be proposed for each of these parts. For the first two parts of the chromosome, a uniform crossover mechanism is used. However, the parents may be infeasible; therefore, we assume that the feasible parents' genes are ten times as dominant. In other words, a gene is ten times more likely to originate from feasible parents than from infeasible parents. For the third and fourth (continuous) parts of the chromosome, an arithmetic crossover function is used.

A two-step mutation process is next applied, in which the first step is general and conducted for all parts of the chromosome. In the first step, the algorithm determines a fraction of the vector entries of an individual to be used for mutation, where each entry has a probability rate of being mutated. According to the results of the settings, this rate is specified as 0.05. In the second step, although the same mechanism is used when the mutation operator is implemented, the two parts of the chromosomes must be distinguished. In this case, the adaptive feasible mutation function is used. The mutation operator chooses a direction and step length that satisfy both the bounds and the linear constraints. After the mutation operator is used, the requirements of the excluded tasks and their task dependencies must be eliminated (set to 0).

To reduce computation time, we specified a multichromosome structure. Since the length of the chromosomes is equal in all runs, the GA can be parallelized. In a multichromosome structure, there are p chromosome vectors. The value of p is usually determined by the number of processors (or graphical coprocessors). The pool contains M multichromosomes in one generation. The processors evaluate these chromosomes in parallel, and selected chromosomes or parts of chromosomes can then be migrated (see Figure 3).

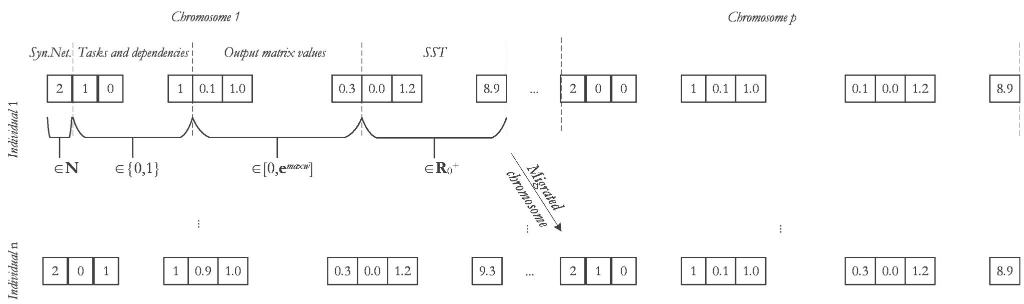


Figure 3: The proposed multichromosome structure

3 Case Study

The implications of our proposed method were evaluated in the R&D sector of a large high-tech automotive company. In this sector, the main target is the provision of software codes for the company's products, which can sometimes consist of millions of lines of code. Software codes are developed iteratively and incrementally following agile approaches. The applied project is an iteration of a real-life software development project that includes nine tasks where all tasks are mandatory apart from one comfort feature. Each feature needs to be implemented and evaluated separately but in parallel based on customer requirements. Software integration must be initiated when all features are evaluated apart from one feature that has a special integration test-relevant part. This task set is generally intended for 4 people and commonly used in almost all software development projects. This project was selected to serve as an example of our method because the executive team includes four people with different DISC personalities. All of these individuals underwent dedicated DISC training where their personalities were identified by a dedicated trainer.

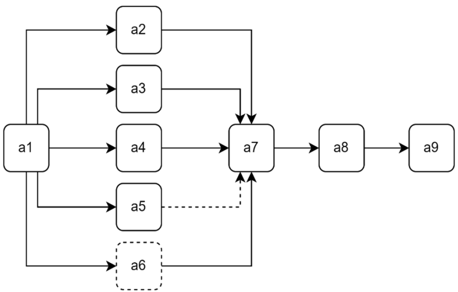
According to Figure 1 and based on the former project results and DISC assessment questionnaire conducted by the trainer, the following synergy matrix was specified. Four team members, whose personality types followed the four DISC personality types, namely, (e1) dominance, (e2) influence, (e3) steadiness, and (e4) conscientiousness were selected. The six soft skills are (s1) leadership ability, (s2) communication ability, (s3) team player attitude, (s4) problem-solving skills, (s5) interpersonal skills, and (s6) analytical thinking. These skills were measured through the use of a 10-point Likert scale to obtain normalized data. The measured soft skills were compared with the competence matrix values that are measured annually by the organization. In this case, we did not see any difference between the two sets of results, so we can conclude that either method can be used for the measurement of soft skills. While soft skills generally depend on personality types, hard skill performances generally depend on former experience. These hard skills represent the implemented functions per week (s7), UX/UI designs per week (s8), UX/UI documented functions and test results per week (s9), and writing deploy and testing scripts per week (s10).

	Y domain				S_s subdomain						S_h subdomain				
	e1 (D)	e2 (I)	e3 (S)	e4 (C)	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	
e1 (D)		1.5	0.5	1.7	1.0	0.8	0.3	0.8	0.8	0.8	4.0	1.5	2.0	1.5	**e1 (D)**
e2 (I)			1.1	0.9	0.8	1.0	1.0	0.6	1.0	0.4	0.0	3.0	1.5	1.0	**e2 (I)**
e3 (S)				1.3	0.5	0.5	0.9	0.5	0.3	0.6	2.5	2.0	3.5	3.0	**e3 (S)**
e4 (C)					0.4	0.5	0.5	1.0	0.4	1.0	5.0	0.0	4.0	1.5	**e4 (C)**
a1	?	?	?	?	1.00	1.00	1.00	0.35	0.83	0.83	4.00	6.50	9.50	4.00	**a1**
a2	?	?	?	?	0.35	0.38	0.60	0.81	0.63	0.75	10.00	6.00	6.00	4.00	**a2**
a3	?	?	?	?	0.42	0.42	0.80	0.77	0.83	0.83	9.2	3.50	8.00	3.50	**a3**
a4	?	?	?	?	0.38	0.35	0.48	0.96	0.79	0.83	10.50	5.50	5.50	2.50	**a4**
a5	?	?	?	?	0.35	0.38	0.56	0.85	0.63	0.79	9.80	6.00	5.50	3.00	**a5**
a6	?	?	?	?	0.31	0.31	0.40	0.73	0.42	0.88	8.00	6.50	4.00	3.50	**a6**
a7	?	?	?	?	0.81	0.85	0.80	0.38	0.83	1.00	3.50	4.50	2.00	6.00	**a7**
a8	?	?	?	?	0.38	0.77	0.36	1.00	0.42	0.83	10.00	6.00	4.00	4.00	**a8**
a9	?	?	?	?	0.96	0.96	0.40	0.35	1.00	0.17	2.00	1.50	9.50	7.00	**a9**
	e1 (D)	e2 (I)	e3 (S)	e4 (C)	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	
	O domain				W_s subdomain						W_h subdomain				

Figure 4: The proposed SMM matrix based on this case study

Since the synergy matrix (Y domain) is symmetric and the diagonal values are 1.0, specifying the upper triangular part of the synergy matrix is sufficient. Figure 4 shows that there is positive synergy between employees who both have D personality types and those between employees with I and those with C personality types, but there are negative synergies between those with D and those with S personality types. Importantly, the specified synergy matrix is in line with the literature, as we saw in the introduction; however, to maintain generality, in the simulations, $\pm$ 20% have been randomly added to every positive and negative synergy value.

The examined sprint has nine tasks specified by its project template: (a_1) the design, (a_2, a_3, a_4, a_5), implementation of functions A, B, C, and D, (a_6) implementation of extra function E, (a_7) testing functions, (a_8) maintenance/improvements, and (a_9) integration . This template is used for most sprints. The priorities are determined using the dynamic system development method (DSDM), where mandatory tasks have relative priority, i.e., a task score of 1. Lower priorities have lower task scores. The flexible dependencies are derived from the technology. The logic network is specified in Figure 5.



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
a_1	1	1	1	1	1	1			
a_2		1					1		
a_3			1				1		
a_4				1			1		
a_5					1		0.5		
a_6						0.7	1		
a_7							1	1	
a_8								1	1
a_9									1

Figure 5: Logic structure of the task precedences

If a leader is not selected for this group, team roles are selected autonomously (O). However, if a leader is selected for a small group, only the synergy of the team leader and the other employees can be dominant; therefore, we obtain four different so-called *dominant synergy networks* (see Figure 6), where only positive or negative synergies are noted.

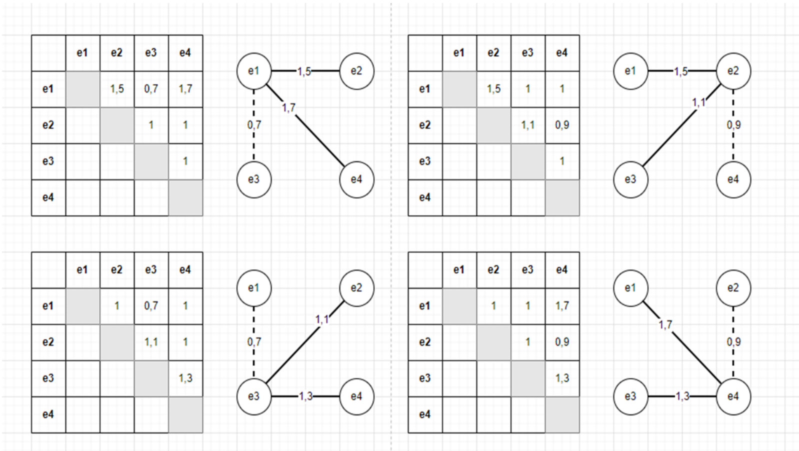


Figure 6: Possible dominant synergy networks

is the example shown in Figure 6 includes only one supplementary task (a_6) and one flexible dependency (a_5, a_7). There is one matching domain, which contains the maximal rates of assignments, in each cell. The simulation specifies five team roles, four target functions, and three relative constraints ($C_x\% \in \{C_s\%, C_t\%, C_c\%\}$). The relative constraints are calculated by minimal and maximal requirements as follows:

$$C_x\% = \frac{TPX_{max} - C_x}{TPX_{max} - TPX_{min}} \tag{6}$$

where $C_x\% \in [\mathrm{TPX_{max}}, \mathrm{TPX_{min}}], \mathrm{TPX} \in \{\mathrm{TPT}, \mathrm{TPC}, \mathrm{TPS}\}$. Since $\mathrm{TPS_{max}} = \infty$ and $\mathrm{TPC_{min}} = 0$, when employees are not assigned to any task, the minimal assignment is specified as half of the maximal assignments ($e^{minw} = \frac{e^{maxw}}{2}$). Thus, $\mathrm{TPT_{max}}$ and $\mathrm{TPC_{min}}$ can be calculated. In this simulation, $C_x\% \in \{0.5, 0.6, \ldots, 1.0\}$. This provides 6^3 kinds of constraint settings. In every setting, we specified 100 simulations to consider the sensitivity of the estimation of synergy values. Therefore, we obtain $6^3 * 5 * 4 * 100 = 432{,}000$ SMM matrixes. According to the case study, the following additional specific subquestions, which are related to the original RQ can be posed:

Sub-RQ_1: Is there any importance to considering synergies?

Sub-RQ_2: According to the constraints and target functions, which team role selection provides the best solutions?

Sub-RQ_3: Which team role selection is the least sensitive to constraint changes?

Descriptive statistics are used to answer Sub-RQ_1, and the effect of accounting for synergy is investigated using Student t tests. Then, as an input to Sub-RQ_2, we examine the effect of the variable that is deemed significant on other variables and constraints through the use of ANOVA. Based on the ANOVA results, whether the groups can be compared according to the significant dependent variables can be determined. Finally, the Bartlett test can be used to answer Sub-RQ_3.

4 Results and discussion

The first Sub-RQ_1, "*Is there any importance to considering synergies?*" is related to the main RQ, "*Which kind of team role selection mechanism provides the shortest project duration considering the constraints?*" and must be answered. Table 2(a) shows the results of the descriptive statistics of the project cost (TPC, 1,000 EUR) and duration (TPT, week), and Table 2(b) shows the results of the pairwise t test between the cases of considering (syn) and neglecting (nosyn) synergies.

Table 2: Comparison of TPT and TPC

	N	Mean	SD	SE
TPT_{syn}	432,000	1.947	1.280	1.947
TPT_{nosyn}	432,000	2.025	1.331	2.025
TPC_{syn}	432,000	78.932	7.588	1.155
TPC_{nosyn}	432,000	83.078	8.079	1.232

(a) Descriptive statistics

Measure 1		Measure 2	t	df	p
TPT_{syn}	-	TPT_{nosyn}	-1.000	431,999	0.317
TPC_{syn}	-	TPC_{nosyn}	-52.409	431,999	< 0.001

(b) Paired samples t test

Table 2(a) shows that the consideration of positive and negative synergies may reduce both the project duration (F) and the TPC. The expected value of the TPT is more than two weeks when these synergies are considered. Considering the synergies between employees can reduce the costs to 4,146 EUR. Nevertheless, Table 2(b) shows that only the difference in project costs is significant. Therefore, only the TPC is examined. The subquestion (Sub RQ_2) "*According to the constraints and target functions, which team role selection provides the best solutions?*" is strongly related to the original RQ. Figure 7 shows the ANOVA results.

Figure 7 shows that only the team role selection variable is significant for the project cost. Neither the target function nor the constraints are significant. It also shows the TPC_{syn} values if the relative cost ($C_c\%$), time ($C_t\%$), and score/scope ($C_s\%$) constraints are neglected.

ANOVA - TPC_{syn}	Sum of Squares	df	Mean Square	F	p
Team role	6,123,182	4	1,534,457	2.655	0.031
Target	1,141,542	3	379,888	0.660	0.577
$C_t\%$	3,254,545	5	649,867	1.129	0.342
$C_c\%$	3,280,458	5	655,003	1.138	0.338
$C_s\%$	2,430,254	5	485,072	0.842	0.519
Residuals	2.49e+11	431,977	575,796		

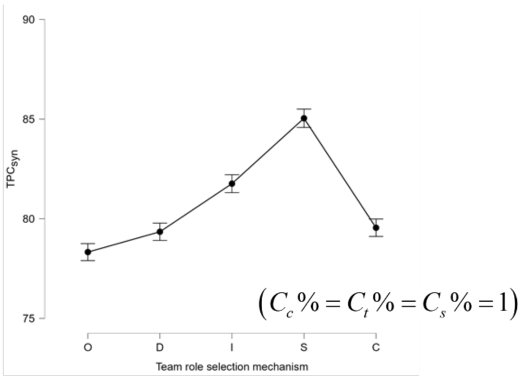


Figure 7: Project costs for the various kinds of team role selection mechanisms, under no constraints.

The lowest cost occurs when autonomous (O) team role selection is employed, while the greatest cost occurs when an employee with a personality type of steadiness (S) is selected to lead the team. Although Table 2 shows that the chancing constraints have no significant effect on the project cost, according to Sub-RQ_3, it is important to answer the question: "*Which team role selection is the less sensitive to the changes of the constraints?*" The Bartlett test shows that the changing constraints have no significant effect on the variance of the project cost. Figure 8 also shows that constraints do not influence the confidence interval of the project cost.

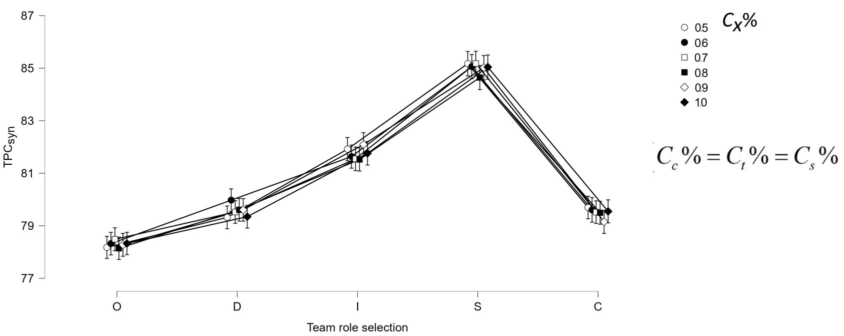


Figure 8: Project costs for the various kinds of team role selection mechanisms, under no constraints

5 Summary and conclusion

In this study, we proposed a modification of the SSPSP problem to distinguish soft skills and hard skill performance. We revealed a method for examining the team role selection mechanism for project time and project cost. To address the modified SSPSP problem, an HGA algorithm with multichromosomes was proposed. The simulations were based on a real-life project, and the case study illustrated how to fill the modified SMM matrix. To answer the RQ, we used a case study to show that autonomous team role selection provides the shortest project cost. Regarding Sub-RQ_1, RQ_2 and RQ_3, we can say that the consideration of synergy affects the project schedule, but only through its costs. (Table 2). We can also state that self-organizing teams are more successful when synergistic effects are considered (Figure 7), but neither is more sensitive to changes in cost (Figure 8).

References

Alba, E., Chicano, J.F., 2007. Software project management with {GAs}. Information Sciences 177, 2380–2401. https://doi.org/10.1016/j.ins.2006.12.020

Al-Saqqa, S., Sawalha, S., AbdelNabi, H., 2020. Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies 14. https://doi.org/10.3991/ijim.v14i11.13269

Antoniou, A., 2019. Compatibility of small team personalities in computer-based tasks. Challenges 10, 29. https://doi.org/10.3390/challe10010029

Arashpour, M., Kamat, V., Bai, Y., Wakefield, R., Abbasi, B., 2018. Optimization modeling of multiskilled resources in prefabrication: Theorizing cost analysis of process integration in off-site construction. Automation in Construction 95, 1–9. https://doi.org/10.1016/j.autcon.2018.07.027

Aslam, W., Ijaz, F., 2018. A quantitative framework for task allocation in distributed agile software development. IEEE Access 6, 15380–15390. https://doi.org/10.1109/ACCESS.2018.2803685

Capretz, L.F., 2003. Personality types in software engineering. International Journal of Human-Computer Studies 58, 207–214. https://doi.org/10.1016/S1071-5819(02)00137-4

Flores-Parra, J.-M., Castañón-Puga, M., Evans, R.D., Rosales-Cisneros, R., Gaxiola-Pacheco, C., 2018. Toward team formation using belbin role types and a social networks analysis approach, in: 2018 IEEE Technology and Engineering Management Conference (TEMSCON). IEEE, pp. 1–6. https://doi.org/10.1109/TEMSCON.2018.8488386

Kosztyán, Z.T., Bogdány, E., Szalkai, I., Kurbucz, M.T., 2022. Impacts of synergies on software project scheduling. Annals of Operations Research 312, 883–908. https://doi.org/10.1007/s10479-021-04467-5

Larson, J.R., Jr., 2009. In search of synergy in small group performance. Psychology Press, New York, US. https://doi.org/10.4324/9780203848784

LePine, J.A., Buckman, B.R., Crawford, E.R., Methot, J.R., 2011. A review of research on personality in teams: Accounting for pathways spanning levels of theory and analysis. Human Resource Management Review 21, 311–330. https://doi.org/10.1016/j.hrmr.2010.10.004

Lykourentzou, I., Antoniou, A., Naudet, Y., Dow, S.P., 2016. Personality matters: Balancing for personality types leads to better outcomes for crowd teams, in: Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing. pp. 260–273. https://doi.org/10.1145/2818048.2819979

Oza, N., Fagerholm, F., Münch, J., 2013. How does Kanban impact communication and collaboration in software engineering teams?, in: 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). IEEE, pp. 125–128. https://doi.org/10.1109/CHASE.2013.6614747

Peeters, M.A., Van Tuijl, H.F., Rutte, C.G., Reymen, I.M., 2006. Personality and team performance: a meta-analysis. European journal of personality 20, 377–396. https://doi.org/10.1002/per.588

Scullard, M., Baum, D., 2015. Everything DiSC Manual. John Wiley & Sons.

Stylianou, C., Andreou, A.S., 2012. A multiobjective genetic algorithm for software development team staffing based on personality types, in: Artificial Intelligence Applications and Innovations: 8th IFIP WG 12.5 International Conference, AIAI 2012, Halkidiki, Greece, September 27-30, 2012, Proceedings, Part I 8. Springer, pp. 37–47. https://doi.org/10.1007/978-3-642-33409-2_5

Vega-Velázquez, M.Á., García-Nájera, A., Cervantes, H., 2018. A survey on the software project scheduling problem. International Journal of Production Economics 202, 145–161. https://doi.org/10.1016/j.ijpe.2018.04.020

Winter, B., 2015. Agile performance improvement: The new synergy of agile and human performance technology. Springer. https://doi.org/10.1007/978-1-4842-0892-2

Wysocki, R.K., 2019. Effective project management: traditional, agile, extreme. John Wiley & Sons. https://doi.org/10.1002/9781119562757

Xiao, J., Ao, X.-T., Tang, Y., 2013. Solving Software Project Scheduling Problems with Ant Colony Optimization. Comput. Oper. Res. 40, 33–46. https://doi.org/10.1016/j.cor.2012.05.007