Wael Almadhoun and Mohammad Hamdan*

Optimizing the Self-Organizing Team Size Using a Genetic Algorithm in Agile Practices

<https://doi.org/10.1515/jisys-2018-0085>

Received February 11, 2018; previously published online December 20, 2018.

Abstract: In agile software processes, the issue of team size is an important one. In this work we look at how to find the optimal, or near optimal, self-organizing team size using a genetic algorithm (GA) which considers team communication efforts. Communication, authority, roles, and learning are the team's performance characteristics. The GA has been developed according to performance characteristics. A survey was used to evaluate the communication weight factors, which were qualitatively assessed and used in the algorithm's objective function. The GA experiments were performed in different stages: each stage results were tested and compared with the previous results. The results show that self-organizing teams of sizes ranged from five to nine members scored more. The model can be improved by adding other team characteristics, i.e. software development efforts and costs.

Keywords: Agile, genetic algorithm, optimization, team size.

2010 Mathematics Subject Classification: 68T04.

1 Introduction

The self-organizing team is at the heart of the agile software development practice [9, 10, 15, 21, 30, 45]. Despite the popularity of agile software development, there has been little research on team size and its relation to a team's performance. There are different characteristics and attributes that can affect the team performance, these are discussed in Section 2. The team communication characteristics has been used in the algorithm design as the objective function. In this research, the genetic algorithm (GA) was modeled based on available theories, previous work and the outcome of survey data to find the optimal team size based on a communication load constraint, as illustrated in Sections 2.3 and 3.1. The typical self-managed team can range from 5 to 15 members [6]. It was found that statistical correlation varies between team size, efforts and productivity [42].

The objective of this research is to develop a GA model to find the optimal self-organizing team size considering team communication characteristics. Section 2 is a literature review, Section 3 is a high-level design methodology and a summary of the users' survey, Section 4 is the algorithm design and implementation, Section 5 is the results and discussion and finally we conclude in Section 6.

2 Literature Review

Most of the work on work-groups' performance, have investigated characteristics associated with effectiveness [34] without considering the effect of those characteristics [8]. The heterogeneity between team members, training, diversity, access to information, self-management, team size, communication,

*Corresponding author: Mohammad Hamdan, School of Mathematical and Computer Sciences, Heriot-Watt University, Dubai, United Arab Emirates; and Department of Computer Science, Yarmouk University, Irbid 21163, Jordan, e-mail: M.Hamdan@hw.ac.uk, hamdan@yu.edu.jo. <https://orcid.org/0000-0001-6477-9111>

Wael Almadhoun: Department of Engineering, ADNOC Offshore, Abu Dhabi, United Arab Emirates

decision-making and experience are possible characteristics [17] and variables that could positively, or negatively, affect the team's productivity. In this section, the self-organizing team, characteristics, communication and a GA are illustrated.

2.1 Self-Organizing Team Definitions

The team is defined as a group who have shared commitments [24] and strive for synergy among the team members. The self-organizing team comes under a social entity work group [56], are self-determined and self-designed, working independently to perform tasks [17, 49]. A group of motivated individuals working as one team together to achieve one goal with the ability and authority to take decisions [20] and ready to respond to high changing demands [3, 15]. Takeuchi and Nonaka [50] have described the team's characteristics as transcendent, autonomous and cross-functional.

In the agile manifesto [1], the self-organizing team is one of the core concepts and one of the 12 principles of the agile software development [9, 10, 15, 30]. It has always been anticipated as people sharing the responsibility of supervisors and teams having everyday decision-making. The team should have variety in terms of multi-discipline, combined skills, redundancy in functions to supplement each other's, also, in learning new skills and having better ways to complete more tasks [22]. The definition of teams and groups, including self-organizing, autonomous, semi-autonomous, self-managing, self-determining, self-designing, crews, cross-functional, quality circles, project teams, task forces and emergency response teams are features of different work groups [17, 21, 35].

2.2 Performance Characteristics

As observed from self-organizing team definitions, the team's productivity can be increased by improving the sense of responsibility and ownership of work [16]. Individuals working collaboratively were found to achieve better decision-making [57]. By eliminating the manager role, the team can handle the decision-making role, but in some cases the team's performance did not improve [51]. Social interdependent teamwork, implies that a cooperative environment between individuals, who are interested in a common goal, sharing skills and knowledge develop mutual relationships [39].

The Karhatsu framework [26, 36, 48] identified the baseline of self-organizing team characteristics which focuses on team orientation, shared leadership, learning, communication and collaboration autonomy. In a research questionnaire [36] which was conducted to evaluate Agile software projects, self-organizing teams were recognized as a performance correlated factor. Generally, the dilemma is, how to measure team performance?

In sports games, the teams' performance can be measured by knowing the winning or losing teams [44], but in software development, there should be other performance measures and factors.

Based on the Hofstede multi-focus model [23], organizational culture is the way members are related to each other, their work, and the outside world. The organizational culture can either enable your strategy or delay it. According to this model, to achieve the optimal culture, to perform gap analysis and to implement the change; Hofstede has studied the following dimensions: Open system, goal-oriented, internally driven, employee-oriented and acceptance of leadership style [23]. It has been perceived by agile experts that having the right corporate culture is a necessary factor to determine the outline of agile practices [10, 56]. As agile methods put a stress on people interactions [19] over processes and on customer collaboration, it is logical that individuals are naturally supportive and collaborative [13, 33].

2.3 Communication and Team Size

The automation in solving problems relates to the team's communication efforts and team size in agile software practice.

2.3.1 Team Size

In the software development industry, researchers have been studying the relation between team size, project efforts and team productivity [5, 18, 38, 42]. In [29], they compared traditional project management and agile practice by considering the requirements prioritization policy with eight other input values, the input values were, the team size and the culture. In addition to dynamism, criticality, interdependency, cost and value, culture was measured in percentage and has a range of 90% in agile projects. Requirements are sorted according to cost effectiveness when completed first. The team size was measured according to number of personnel and it has a range of 3 (1, 8, 20). The performance score considered the work to be done using the final value/cost after project completion. By modeling previous values and using an AI search engine in a distribution of 10,000 runs [29], the results showed that, the agile method provided better performance than the plan-based methods. For very high dynamism, it was found useful to have a team size of five and not more than 17 people.

The software development efforts and the impact of the team size in addition to other variables have been investigated in over 200 projects as part of an empirical study [37]. The variables included: software size in terms of functional requirements, programming language and management tools. The larger team size increases the software development efforts due to high interaction, coordination and communication between the team members [38]. One hypothesis is that, the team size has a positive association with the software development efforts. Team interaction with tools has no effect and the interaction of team size and programming language has a negative association are other hypotheses. However, using ordinary least squares (OLS) analysis and using data quality to measure the hypotheses [37], the results found no support for the previously mentioned hypotheses.

2.3.2 Communication

With the complexity of software projects, the emerging technologies and the multiple frameworks available to developers, it became fundamental to build multi-skilled teams. However, it is important that team members can effectively communicate with each other, with other teams and with their leaders [14, 31, 40]. People interacting with each other in the form of communication, will be used in the fitness function as part of the GA, which aims to find the optimal team size.

In real-world communications, expertise is as important as the technical skills required within the team. It is also possible to study the communication constraint in a social network to find the optimal team with minimum communication cost [52]. Two communication metrics were introduced to measure the team's communication effectiveness: (1) diameter communication cost and (2) minimum spanning tree cost. Given a group of experts $X = \{x_1, x_2, x_3, \dots, x_n\}$, and a list of skills $S = \{s_1, s_2, s_3, \dots, s_n\}$. The social network was modeled using an undirected graph $G(X, E)$ where E are edges. Weight of (x_i, x_j) is the communication cost in project $P \subseteq S$, a team $T \subseteq X$, so that $s \in \text{skills}(x)$.

The algorithm brute force was used to solve the team formation problem with the communication load constraint, which includes two phases. The first phase is team generation, and the second phase is hierarchy establishment with minimum communication cost for each generated team in phase one. The algorithm shows slow performance when the network nodes were increased from 30 to 300. The opt-algorithm was used to overcome the performance challenge which was found satisfactory [52]. The skills required for each team is set to 6 and the communication load constraint is set to 3. According to the results, the opt-algorithm was found to be the best of the three algorithms. However, the research did not mention the optimal team size considering communication load constraints.

A different interaction requires a different formula to calculate the number of communication channels of each type. In this project, the focus is on self-organizing team which requires all channels communications type. The following is a list of different communication channels types as shown in Figure 1:

- Wheel: all communication goes through one the group leader.
- Y-Pattern: less centralized – two persons are closer to the center.
- Chain: flow of information goes among members from one to one till the end.

Figure 1: Different Patterns of Communication [27].

Figure 2: The Relation Between the Team Size and the Number of Communication Channels is Exponential. Increasing the number of team members will exponentially increase the possible number of interactions. This means more members in the team, requires more communication channels [7, 25].

- Circle: closer sides persons can communicate with each other.
- All channel: more decentralized, communication flows freely between all team members.

All channel (star) type of communication is considered in the GA design. As shown in Figure 2, increasing the number of team members will exponentially increase the number of communication channels.

2.4 Genetic Algorithm (GA)

GAs are inspired by natural biological evolution which tends to find the best solution to solve a problem. The GA was invented by John Holland in 1975 and was popularized by Goldberg in 1989 [47, 53, 54]. The GA is known as evolutionary algorithm for optimization using techniques inspired by nature [55]. This means that the algorithm tries to find the closest best solution of the available solutions, but is not guaranteed to find the optimal one [12]. A population of solution candidates solve the optimization problem and move towards the best solution [11]. The algorithm consists of three generic operators, which are applied on population generations (selection, crossover, and mutation) [28]. The fitness function is being evaluated for each individual in the population, to select individuals for reproduction through crossover and mutation, to create the next generation [54].

The solution space is required to find an optimal parameter value which can be represented by a binary string or a real number. For example, the team size can be represented by the values 3, 5, or 10 members. But consider that the team member might be present or virtually connected, which can be represented by

different parameters, or by a 10-bits binary string, for example, in [11010 | 11011] the first 5-bits represents the team size value and the second 5-bits represents the team type [12].

Fitness Function: Once the solution is represented by encoding strings. It is necessary to evaluate the solution to meet the principal objective. In most cases, the fitness function is equal to the objective function. For example, the following fitness function is used to calculate the number of communication channels available for a team of five persons:

$$f_s = \frac{n(n-1)}{2} \quad (1)$$

The fitness function f_s used the team size $n = 5$ to calculate the total number of communication channels, which was found to be 10. In this example, looking for optimized solution would be minimizing the fitness value. The flowchart in Figure 3 presents the GA steps.

Figure 4 illustrates the crossover process and Figure 5 represents a mutation operation in bit number 5.

Figure 3: Working Principle of the GA [12].

The steps involved in implementing the algorithm are inspired from nature. First, the chromosome encoding is used to represent the solution. Then, a random population of individuals is generated. Then, the fitness value is used to determine the set of values that best fit for selection. In the selection stage, the algorithm picks the parents that are stronger to survive. Then, the reproduction stage, the stronger parents have higher probability to produce better children's using crossover. Then, the mutation stage is used to modify and diversify the new population.

Figure 4: Crossover: Generally, the Crossover is Used to Create a New Solution from the Existing Solution After Applying the Selection.

It can be done by exchanging two chromosomes in a crossover point [12, 53].

Child	1	0	0	1	1	0	1	0	1	1
Child _{mutated}	1	0	0	1	0	0	1	0	1	1

Figure 5: Mutation: Mutation is Optional Step, Which is the Transformation of Randomly Selected Genes.

Mutation helps to occasionally introduce new features to maintain diversity and avoid local optima. A common mutation rate is 5%.

Chromosome Encoding: The chromosome is the raw genetic information that the GA deals with (genotype). The physical representation of the chromosome is known as (phenotype) [47]. A chromosome is a string which stores individual genetic information. The string length depends on number of genes and encoding type [43].

3 Design Methodology

In this section, the genetic algorithm solution requirements are outlined briefly as a prerequisite for the design and implementation stage. First, the number of communication channels between individuals and teams is the objective function. Then, other constraints were added to the communication factor to calculate the fitness function. Then different categories and medium of communication channels were discussed. The details of GA design, equations and pseudocode are covered later in Section 4.

3.1 Communication Channels Calculation

The following formula has been used to calculate the number of communication channels, as per the PMI best practices [25]. Assume that, N is the total number of communication channels, n is the number of team members for each team and $T = \{t_1, t_2, \dots, t_n\}$ is the number of teams. The following equation was used to calculate N_t the total number of communication channels within the same team:

$$N_t = \frac{n(n-1)}{2} \quad (2)$$

To calculate N_T the total number of communication channels between teams:

$$N_T = \frac{T(T-1)}{2} \quad (3)$$

To calculate N the total number of communication channels:

$$N = T \times N_t + N_T \equiv \quad (4)$$

$$N = \frac{T(n(n-1) + (T-1))}{2} \quad (5)$$

For example, assume a project team of 15 persons: Total number of communication channels = $(15 \times 14)/2 = 105$. Now, if the team has been organized to three small teams of five: The total number of communication channels $N = 3(5(5-1) + (3-1))/2 = 33$.

The GA fitness function was designed based on the previous communication channels formula, but considering other attributes and different team models.

Communication Weight

In the previous equations, it was considered that all communication channels have the same weight. But considering that, they have different effectiveness, the weight factors will be added in the algorithm design to calculate the fitness value as listed in Table 1. Based on the effectiveness of the communication channel shown in Figure 6, the different communication types are having different weight factors. These weight

Table 1: Communication Weight Factors.

Description	Weight factor	Value
Verbal person to person communicating	w_1	1
Team to team communication	w_2	1.514
Person to person communicating by email	w_3	1.264
Person to person communicating via video conference	w_4	1.266
Person to person communicating in a second language	w_5	1.362

Figure 6: Communication Options [2].

The direct verbal/oral communication is used as a reference for comparison with other types of communication channel's medium. Considering that, the video conferencing is usually used with virtual teams [31]. The effect of using different communication channels were considered.

factors were quantified and used in the objective function to calculate the communication efforts. It was gathered from the survey data, as shown in Section 3.2. Figure 7 illustrates a sample population of size 100, in the histogram the distribution of individuals represents the team size and the frequency represents how many times repeated in the solution population. Figure 8 is a high level representation for the communication

Figure 7: The Chart Shows the Histogram Plot for a Small Population. The popular range of the solution space can be set between 100 to 400.

Figure 8: The Node Network is the (phenotype) Solution Representation.

The tuple of all possible node to node communications were created, e.g. nodes = [(1, 2), (1, 3), (2, 6), (4, 6), (4, 5), (5, 6), (1, 4), (1, 5), (1, 6), (2, 3), (3, 6), (2, 5), (3, 4), (2, 4), (3, 5)].

channels within the same team. To fully illustrate the star type (all channel) communication, the cross team communications should be plotted, but it would be difficult to visualize.

3.2 Survey and Data Analysis

A survey was conducted to quantify the communication constraints as discussed in Section 3.1. Out of total 78 participants, 56 of them have worked in agile projects.

Considering that “person to person communicating in verbal” is having the weight factor of 1, which was used as a reference for the highest communication channel effectiveness, so that, the higher weight factor means lower effectiveness. The following are some of the important questions:

- (I) Q. person to person email or chat messaging communication has a weight factor of? Thirty-eight percent of participants considered that email communication is the same as verbal communication and 28% considered them close.
- (II) Q. person to person from different cultural backgrounds or communicating in a second language has a weight factor of? Seventy-seven percent of participants believe that communicating in second language requires more effort.
- (III) Q. person to person communicating in a video conference has a weight factor of? Twenty-seven percent believed that more effort is required.
- (IV) Q. person female to person male verbal communication has a weight factor of? Fifty-six percent believed that weight factor for both genders is the same.
- (V) Q. compared to a person’s communication; team to team communication has a weight factor of? Sixty-six percent of the participants believed that, more effort is required to communicate across teams.

4 Design and Implementation

As mentioned in the literature review in Section 2, there are a number of steps involved in the GA operation. These steps are: the randomly generated population, fitness function, selection, reproduction and termination. The code was developed using *Python* scripting language, which facilitated the testing of different models and scenarios.

4.1 Genetic Algorithm Operators

Population: The randomly generated population represents the solution space, which contains the initial possible individual solutions, encoded as detailed below. The population size can vary and has big influence on the GA results. A too small population size might not provide the desired results and a too large population size does not necessarily improves them. As per Grefenstette, the GA best population size should be between 10 and 160 [53].

4.1.1 Chromosome Encoding

The string value was used to store the individual's information.

Genotype: The solution for different team sizes has a discrete value, so the real number of chromosome encoding was considered. As was covered in the literature review, the value of each team can range from 3 to 17 members. Below is a sample of randomly generated strings that represent a distribution of total 120 members into small self-organizing teams: Gene = [6, 6, 10, 3, 8, 10, 4, 7, 7, 4, 6, 6, 5, 11, 9, 11, 7].

Phenotype: The phenotype was used to physically represent the solution, as the possible number of communication channels for each team. Each team has a several nodes and each member represents a node.

Fitness Function: The objective function for calculating the number of communication channels and the communication load constraints was used as the fitness function. The individuals with the best fitness values are selected to generate the next generation. The GA process shall be working towards minimizing the fitness values.

Assume, the person's communication means member to member communication, and the team communication means team to team communication. The email communication and communicating in a second language are related to persons. Also assume, that using the video conference is related to the virtual team.

The total team size of N is distributed to smaller teams of t , where n is the team internal number of communication channels, w_1 is the person to person communication weight factor and w_2 is the team to team communication weight factor. The fitness function f_t :

$$persons_{weight} = w_1 \times w_3 \times w_5 \quad (6)$$

$$team_{weight} = w_2 \times w_4 \quad (7)$$

Algorithm 1: Simplified GA Structure.

```

1: set  $i = 0$ 
2: set maximum generations  $\gamma = 10,000$ 
3: initialization ( $I: \mu \rightarrow POP$ ) coding function ( $\varphi: \Sigma \rightarrow G$ )
4: GetFitness  $f_t()$ 
5: Crossover  $Cx()$ , crossover rate  $PCx$ 
6: Mutation  $Mx()$ , mutation rate  $PMx$ 
7: loop:
8:   if  $i \leq \gamma$  then
9:      $PopFitness \leftarrow f_t(POP)$ .
10:    Generate new population ( $R: \varphi \times PCx \rightarrow POP$ )
11:     $G_a \leftarrow Cx$ .
12:     $POP \leftarrow sorted(G_a) \leftarrow Mx$ .
13:    goto loop.
14:   end if
15: Solution =  $min(POP)$ 

```

$$f_t = \sum_0^t \left(\frac{n(n-1)}{2} \right) * persons_{weight} + \left(\frac{N(N-1)}{2} \right) * team_{weight} \quad (8)$$

4.2 Genetic Algorithm Pseudo-Code and Parameters

Using the following parameters: $\sum$ represents the search space, μ is the population size, POP is the population of individuals, Ga is the new generation population, γ is the maximum number of generations. The GA pseudo-code is shown in Algorithm 1.

5 Results

5.1 Teams are Equally Distributed

The self-organizing team tends to be multi-disciplinary. One team of five members could have a combination of one analyst, one tester and three developers. The following given teams were tested considering that, the teams will be equally distributed. Each team of the following list was examined by the GA (*Test 1*): Total team members [50, 100, 120, 150, 200]. The total number of project members depends on the project size. The research study is to find the optimal self-organized team distributions. For a team of total number below 50, it can be solved without the GA. There was no reference from the software industry of a team bigger than 200 member. The first test results shown in Table 2.

Test 1 Results:

Outcome: The most optimal team size results for self-organizing teams that are equally distributed were found to be five, seven, or nine members per team depending on the given total team size and the following notes.

Interpretation: the team size was found in range between five to nine members per team. The recommended team size by agile best practice is seven plus/minus two members.

Discussion:

- Considering the human factor, the team size is a discrete value. Some teams can not be equally divided; therefore, the rounded figure was used in the GA to calculate the team distribution. For example, the 100 members test result shows that the team can be divided into 14 teams, each has seven members. The result has 2% error, as two team members were not assigned.
- As part of the results, the team sizes with six or eight are even numbers. The agile practitioners did not recommend teams of even numbers, due to equal voting possibilities in case of disputes and decision-making or in the case of assigning the team to a leader, then the team even number is chosen [41].
- The test results did not improve after the GA reproduction stage in the second generation. By reviewing the results in (Table 2), it was found that all the possible answers did exist in the first population. This is due to the small solution space.

Table 2: Test-1 Cases and Results.

Total team members	Fitness value	Optimal team size
50	162.00	6
70	276.50	5
100	430.50	7
120	561.00	7
150	756.00	7
200	1138.50	9

5.2 Teams are Randomly Distributed

In this section, the team distribution was unequally and randomly generated, so that the decision will be taken by the GA based on the total team fitness value. The case of having equal teams might not be the optimal case? The human mind might not imagine the optimal random distribution of a team and the relation to its communication load constraint.

Test 2 Results:

The test cases results are shown in Table 3. In the previous test, the results were shown as a single value, so the total team members will be equally distributed into an unique value, e.g. 5. In this test, the results are different, they were represented as a set of values that its total sum is equal to the total number of team members, e.g. [5, 6, 6, 6, 7]. The results of team distributions are shown as pie chart plots in Table 4.

Table 3: Test-2 Cases and Results.

Total team size	Team distribution
50	[5, 5, 5, 6, 6, 6, 5, 6, 6] Has fitness value of: 169.0
70	[6, 5, 6, 6, 6, 6, 6, 5, 5, 7, 6, 6] Has fitness value: 270.0
100	[7, 9, 7, 6, 7, 6, 6, 7, 6, 6, 7, 7, 5, 7] Has fitness value: 446.5
150	[9, 6, 7, 7, 8, 7, 9, 7, 9, 7, 6, 8, 9, 6, 8, 8, 8, 7, 7, 7] Has fitness value: 782.0
170	[8, 8, 9, 9, 8, 6, 7, 9, 9, 6, 7, 9, 9, 9, 7, 9, 7, 8, 8, 9, 9] Has fitness value: 929.0
200	[8, 10, 7, 8, 8, 8, 8, 7, 9, 9, 10, 9, 9, 9, 7, 11, 10, 8, 10, 9, 8, 8, 10] Has fitness value: 1162.5

Table 4: Team Distribution Pie-chart.

Figure 9: Communication Nodes and Channels for a Team of 100 Members.

Figure 10: Histogram-Sample Population for a Team Distribution of 200 Members.

Each test has several plots. Due to the large number and size of the plots, only sample plots are included. The plot in Figure 9 for a team distribution of 100 members of *phenotype* with team to team communication channels are connected. Figure 10 shows a sample initial population for a team with a total of 200 members, which represents the diversity of individuals in the first generation. Figure 11 shows the fitness function minimization during the test of a team size of 120 members.

5.3 Evaluation

This project aims to find the optimal team size and distribution for a given total number of team members. In the test results, it was found that the team size can range from five to nine members per team, which matches the industry's best practice. The following researches results were found to support this project evaluation:

- In the research of “an approach to optimizing software development team size” based on a 4000 project data analysis [18], the software cost estimate was considered and a small team was recommended. The best team size was found to be seven.

Figure 11: Testing a Team of 120 Members.
The fitness function minimized over generations.

- In an empirical study, for the impact of team size on software development efforts. The team size of nine was found to be productive [37].
- Researchers have rewarded team size in high performance teams. It was suggested that team members should not exceed 12, otherwise subgroups shall be considered to provide a cover of anonymity [32].
- The research by [29] has referenced a NASA recommendation for the team size with a wide range from three to 19 member per team.
- In the “Experimental approach to organization development” [6] the small team size was mentioned as a characteristic of the self-managed team which can range from five to 15 member per team.

6 Conclusion and Future Work

In this research, the study of team productivity was limited to team size and communication characteristics. The number of communication channels between team members and between teams has been used to calculate the communication load constraint. This calculation was used initially as a fitness function by the GA to find the optimal team size from a randomly generated solution space.

The experimental results concluded that, the teams of five to nine members are found to be optimal in different cases. In some cases, equally distributed teams have provided better results than randomly distributed teams. For small to medium projects of 70 members or less, it was found that smaller self-organizing teams of five, six and seven members have better valuation than other team sizes. But, for larger projects which have more than 120 members, it was found that larger self-organizing team of seven, eight and nine members have provided better results than other team sizes.

Following this research results, the proposed GA model will open the door for future work and improvements. In this research, manual input was used with few parameters assuming that each organization will have different situations. Therefore, it would be more productive to build an adaptive learning algorithm and expand the predictability of self-organizing team productivity based on team size and communication. Additionally we can include more characteristics.

With reference to self-organizing team characteristics which are illustrated in Section 2, it is worth studying additional characteristics other than communication efforts. This might include but not be limited to users’ experience, skills, and roles. This could lead to improved fitness function and better results. Some researchers have studied the effect of project scope and the functional requirements development efforts. Also, some researchers used the software size to estimate the required efforts and time, based on the COCOMO approach [4, 46]. There is a possible integration by adding the team size to these estimates. Also, revising the fitness function to find the optimal team size based on the software development cost estimate.

The GA model can be improved by performing sensitivity analysis to optimize the decision variables, also, by using different encoding techniques, if additional decision variables are considered.

Bibliography

- [1] Agilemanifesto, <http://agilemanifesto.org/> (2017), [Online; accessed 21 March 2017].
- [2] S. W. Ambler, Best practices for agile lean documentation, <http://www.agilemodeling.com/essays/agileDocumentationBestPractices.htm> (2012), [Online; accessed 02 July 2017]
- [3] B. Boehm, Get ready for agile methods, with care, *Computer* **35** (2002), 64–69.
- [4] B. Boehm, C. Abts and S. Chulani, Software development cost estimation approaches a survey, *Ann. Software Eng.* **10** (2000), 177–205.
- [5] E. Bradner, G. Mark and T. D. Hertel, Team size and technology fit: participation, awareness, and rapport in distributed teams, *IEEE Trans. Professional Commun.* **48** (2005), 68–77.
- [6] D. R. Brown and D. F. Harvey, *An experiential approach to organization development*, Pearson Education, New York, 2011.
- [7] B. S. Butler, Membership size, communication activity, and sustainability: a resource-based model of online social structures, *Inform. Syst. Res.* **12** (2001), 346–362.
- [8] M. A. Campion, G. J. Medsker and A. C. Higgs, Relations between work group characteristics and effectiveness: implications for designing effective work groups, *Personnel Psychol.* **46** (1993), 823–847.
- [9] T. Chow and D. B. Cao, A survey study of critical success factors in agile software projects, *J. Syst. Software* **81** (2008), 961–971.
- [10] A. Cockburn and J. Highsmith, Agile software development, the people factor, *Computer* **34** (2001), 131–133.
- [11] M. Compare, F. Martini and E. Zio, Genetic algorithms for condition-based maintenance optimization under uncertainty, *Eur. J. Operational Res.* **244** (2015), 611–623.
- [12] K. Deb, An introduction to genetic algorithms, *Sadhana* **24** (1999), 293–315.
- [13] P. Dourish and V. Bellotti, Awareness and coordination in shared workspaces, In: *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, pp. 107–114, ACM, Toronto, ON, Canada, 1992.
- [14] J. A. Espinosa, S. A. Slaughter, R. E. Kraut and J. D. Herbsleb, Familiarity, complexity, and team performance in geographically distributed software development, *Organ. Sci.* **18** (2007), 613–630.
- [15] M. Fowler and J. Highsmith, The agile manifesto, *Software Development* **9** (2001), 28–35.
- [16] D. Goleman, *Leadership that gets results* (Harvard Business Review Classics), Harvard Business Press, Cambridge, MA, 2017.
- [17] R. A. Guzzo and M. W. Dickson, Teams in organizations: recent research on performance and effectiveness, *Ann. Rev. Psychology* **47** (1996), 307–338.
- [18] M. Hericko, A. Zivkovic and I. Rozman, An approach to optimizing software development team size, *Inform. Process. Letters* **108** (2008), 101–106.
- [19] J. R. Highsmith, *Agile project management: creating innovative products*, Pearson Education, London, 2009.
- [20] J. Highsmith and A. Cockburn, Agile software development: the business of innovation, *Computer* **34** (2001), 120–127.
- [21] R. Hoda and L. K. Murugesan, Multi-level agile project management challenges: a self-organizing team perspective, *J. Syst. Software* **117** (2016), 245–257.
- [22] R. Hoda, J. Noble and S. Marshall, Self-organizing roles on agile software development teams, *Software Eng. IEEE Trans.* **39** (2013), 422–444.
- [23] Hofstede, <https://geert-hofstede.com/cultural-dimensions.html> (2000), [Online; accessed 21 March 2017].
- [24] D. R. Ilgen, J. R. Hollenbeck, M. Johnson and D. Jundt, Teams in organizations: from input-process-output models to imoi models, *Annu. Rev. Psychol.* **56** (2005), 517–543.
- [25] G. Indelicato, *A guide to the project management body of knowledge (PMBOK® guide)*. Project Management Journal, **40** (2009) 104.
- [26] H. Karhatsu, M. Ikonen, P. Kettunen, F. Fagerholm and P. Abrahamsson, Building blocks for self-organizing software development teams a framework model and empirical pilot study. in: *Software Technology and Engineering (ICSTE), 2010 2nd International Conference*, vol. 1, pp. V1–297, IEEE, Puerto Rico, USA, 2010.
- [27] Kkhsou: <http://kkhsou.in/main/english/communication.html> (2011), [Online; accessed 29 June 2017].
- [28] A. Kumar, R. Nagar and A. S. Baghel, A genetic algorithm approach to release planning in agile environment, in: *Information Systems and Computer Networks (ISCON), 2014 International Conference*, pp. 118–122, IEEE, Mathura, India, 2014.
- [29] B. Lemon, A. Riesbeck, T. Menzies, J. Price, J. D'Alessandro, R. Carlsson, T. Prifiti, F. Peters, H. Lu and D. Port, Applications of simulation and ai search: assessing the relative merits of agile vs traditional software development, in: *Automated Software Engineering, 2009. ASE'09. 24th IEEE/ACM International Conference*, pp. 580–584, IEEE, Auckland, New Zealand, 2009.
- [30] R. C. Martin, *Agile software development: principles, patterns, and practices*, Prentice Hall, Upper Saddle River, NJ, 2002.

- [31] L. L. Martins, L. L. Gilson and M. T. Maynard, Virtual teams: what do we know and where do we go from here? *J. Manage.* **30** (2004), 805–835.
- [32] E. McKenna, *Business psychology and organizational behaviour*, 5th ed., Psychology Press, Hove, 2012, ISBN-13: 978-1848720350.
- [33] S. C. Misra, V. Kumar and U. Kumar, Identifying some important success factors in adopting agile software development practices, *J. Syst. Software* **82** (2009), 1869–1890.
- [34] N. B. Moe, T. Dingsøyr and T. Dybå, Understanding self-organizing teams in agile software development, in: *Software Engineering, 2008. ASWEC 2008. 19th Australian Conference*, pp. 76–85, IEEE, Perth, Australia, 2008.
- [35] S. Nerur and V. Balijepally, Theoretical reflections on agile development methodologies, *Commun. ACM* **50** (2007), 79–83.
- [36] V. Oza, P. Kettunen, P. Abrahamsson and J. Munch, Attaining high-performing software teams with agile and lean practices: an empirical case study, arXiv preprint arXiv:1311.6933 (2013).
- [37] P. Pendharkar and J. Rodger, An empirical study of the impact of team size on software development effort, *Inform. Technol. Manag.* **8** (2007), 253–262.
- [38] P. C. Pendharkar and J. A. Rodger, The relationship between software development team size and software development cost, *Commun. ACM* **52** (2009), 141–144.
- [39] T. Pina and L. Joe, Employees, teamwork and social interdependence – a formula for successful business? *Team Perform. Manag.* **8** (2002), 54–59.
- [40] M. B. Pinto and J. K. Pinto, Project team communication and cross-functional cooperation in new program development, *J. Product Innovation Manag.* **7** (1990), 200–212.
- [41] A. Qumer and B. Henderson-Sellers, A framework to support the evaluation, adoption and improvement of agile methods in practice, *J. Syst. Software* **81** (2008), 1899–1919.
- [42] D. Rodriguez, M. A. Sicilia, E. Garcia and R. Harrison, Empirical findings on team size and productivity in software development, *J. Syst. Software* **85** (2012), 562–570.
- [43] F. Rothlauf, *Representations for genetic and evolutionary algorithms. representations for genetic and evolutionary algorithms*, 2nd ed., Springer, Berlin, Heidelberg, 2006.
- [44] B. Senior, Team roles and team performance: is there really a link? *J. Occup. Organ. Psychology* **70** (1997), 241–258.
- [45] H. Sharp and H. Robinson, An ethnographic study of xp practice, *Empirical Software Eng.* **9** (2004), 353–375.
- [46] A. F. Sheta, Estimation of the cocomo model parameters using genetic algorithms for NASA software projects, *J. Comp. Sci.* **2** (2006), 118–123.
- [47] S. N. Sivanandam, *Introduction to genetic algorithms*, Springer, Berlin, Heidelberg, 2007.
- [48] P. Srivastava and S. Jain, A leadership framework for distributed self-organized scrum teams. *Team Perform. Manag.: Internat. J.* **23** (2017), 293–314.
- [49] J. Sutherland, A. Viktorov, J. Blount and N. Puntikov, Distributed scrum: agile project management with outsourced development teams, in: *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference*, pp. 274a–274a, IEEE, Waikoloa, HI, USA, 2007.
- [50] H. Takeuchi and I. Nonaka, The new product development game, *Harvard Bus. Rev.* **64** (1986), 137–146.
- [51] J. Tata and S. Prasad, Team self-management, organizational structure, and judgments of team effectiveness, *J. Managerial Issues* **16** (2004), 248–265.
- [52] Y. C. Teng, J. Z. Wang and J. L. Huang, Team formation with the communication load constraint in social networks, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 125–136, Springer, Berlin, 2014.
- [53] M. Unal, Optimization of PID controllers using ant colony and genetic algorithms. *Studies in computational intelligence*, vol. 449, Springer, Berlin, Heidelberg, 2012.
- [54] M. D. Vose, *The simple genetic algorithm: foundations and theory*, vol. 12, MIT press, Cambridge, MA, 1999.
- [55] D. Whitley, A genetic algorithm tutorial, *Statist. Computing* **4** (1994), 65–85.
- [56] E. Whitworth and R. Biddle, The social nature of agile teams, in: *Agile Conference (AGILE)*, 2007, pp. 26–36, IEEE, 2007.
- [57] C. W. Wong, K. h. Lai, T. Cheng and Y. V. Lun, The role of it-enabled collaborative decision making in inter-organizational information integration to improve customer service performance, *Internat. J. Prod. Econ.* **159** (2015), 56–65.