Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.015657
Article

Tech Science Press

Selection and Optimization of Software Development Life Cycles Using a Genetic Algorithm

Fatimah O. Albalawi and Mashael S. Maashi*

Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, 11451, KSA
*Corresponding Author: Mashael S. Maashi. Email: mmaashi@ksu.edu.sa
Received: 01 December 2020; Accepted: 17 January 2021

Abstract: In the software field, a large number of projects fail, and billions of dollars are spent on these failed projects. Many software projects are also produced with poor quality or they do not exactly meet customers' expectations. Moreover, these projects may exceed project budget and/or time. The complexity of managing software development projects and the poor selection of software development life cycle (SDLC) models are among the top reasons for such failure. Various SDLC models are available, but no model is considered the best or worst. In this work, we propose a new methodology that solves the SDLC optimization problem using a genetic algorithm. The methodology selects the best SDLC and optimizes the completion time of the selected model. This study aims to help project managers in a software development organization select the proper SDLC model for their projects and optimize the selected model by minimizing the project completion time. The proposed SDLC model selection approach is based on a selection matrix that consists of a set of selection criteria and information related to the project's nature given by the project managers. Our methodology optimizes the selected SDLC model by reducing the project completion time and assigning duration for each phase. Several experiments were conducted to obtain the optimal completion time for the selected SDLC models. These experiments showed that our algorithm can optimally minimize the completion time of a given project and assign a duration for each phase. Experimental results showed that our methodology can reduce the completion time of a given project and produce realistic and optimal completion times for different SDLC models.

Keywords: Genetic algorithm; software development process; software development life cycle; SDLC; software project manager; optimization; evolutionary algorithms; search-based software engineering; SBSE; optimization problems

1 Introduction

Software development life cycle (SDLC) is a process used in software engineering (SE) [1]. It consists of a set of finite phases and activities that will be conducted during software development [2]. SDLC models define a complete plan for producing high-quality and reliable software products within time and budget targets. They describe how to plan, analyze, design, implement, and test software projects [1,2]. Various

SDLC models are available in the scientific literature, but no model is considered the best or worst. Each model has advantages and drawbacks. No specific model is suitable for all project types [3]. The most popular SDLC models in the literature are the waterfall model, iterative and incremental development (IID) model, spiral model, agile model, rapid application development (RAD) model, and software prototyping [2–5]. Many studies have discussed the advantages and drawbacks of SDLC models [1–5]. In software project management, selecting the best SDLC model is one of the critical issues. Managing software projects is a complex task [3]. The software project manager has a vital role in the project's success, and they should select an appropriate SDLC model. The project manager needs guidance in selecting the best SDLC model for the project. The project manager has to deal with challenges during software development to deliver the project within time and budget. This is due to the complexity of managing software projects, which demand complex management involving planning and scheduling [3]. In project management, the human resources and tasks should be efficiently controlled to achieve a specific objective, such as minimizing cost and time or maximizing profit and productivity. Reducing the software completion time becomes an essential requirement for the early introduction of the software to the market to gain the benefits of market share and profitability [6]. Minimizing the project completion time optimally is known as the "search problem" or "optimization problem." This type of problem can be solved using search-based SE (SBSE) techniques. The project manager needs additional help to solve this problem.

The concept of SBSE was presented by Herman and Jones [7]. The term "search" refers to metaheuristic search-based optimization techniques [7,8]. SBSE is related to the areas of SE and uses search-based optimization techniques to solve different SE problems as optimization problems. Search-based optimization techniques are widely applied to solve many optimization problems in SE, such as problems related to software requirement engineering, testing, design, refactoring, and management. SBSE aims to identify optimal solutions to a given problem in a search space of candidate solutions. To solve a given problem, the problem representation should clearly define an objective function. Then, the solutions are evaluated on the basis of their quality (worse and better) during the search process [8,9]. Many search-based techniques have been proposed to solve various SE optimization problems. The most widely used are genetic algorithms (GAs), simulated annealing (SA), particle swarm optimization, ant colony optimization, and hill climbing (HC) [7]. All of these algorithms are known as evolutionary algorithms (EAs).

In this study, we address two main issues: (i) To enable the project manager to select the best SDLC model, how can we define a mechanism for SDLC model selection, and which criteria can be used? (ii) To enable the project manager to optimize the SDLC model selected, how can we optimize this model to achieve the optimization objective (minimizing the project completion time and assigning duration for each phase), and which search-based technique can be used to solve this problem to provide an optimal solution? To solve these issues, we have to meet the following objectives: (i) Help the user (project manager) select the best SDLC model. The SDLC model selection is in accordance with the information related to project characteristics given by the user and based on a set of selection criteria and matrices. (ii) Optimize the SDLC model selected with the objective of minimizing the project completion time and assigning an optimal duration for each phase. This optimization problem can be solved by applying an SBSE technique to provide an optimal/near-optimal solution. In this study, we propose a methodology for SDLC model selection and optimization. The aims are to help project managers select the best SDLC model and optimize the model selected by reducing the completion time of the project and assigning duration for each phase.

The major contributions of this study are as follows:

- We propose a selection mechanism based on a selection matrix for the most frequently used SDLC models.
- We design a framework for SDLC selection–optimization.
- We formulate an SDLC optimization problem.

- We apply a GA to solve our SDLC optimization problem and design a GA's crossover operator (crossover points) to apply to our problem.

The rest of this article is structured as follows. Section 2 provides the background and related works. Section 3 presents the proposed SDLC selection–optimization methodology. Section 4 describes the experimental settings. Section 5 presents the results and discussions. Section 6 provides the conclusion and suggestions for further research.

2 Background and Related Works

GAs are a type of EAs introduced in [10]. GAs have been successfully applied to solve complex optimization problems. The main reasons for the success of GAs are their ease of use, flexibility, accuracy, and broad applicability. Therefore, we apply a GA to solve our problem [11]. In general, a GA starts with generating an initial population. A population is a random set of initial candidate solutions or individuals that satisfy the constraints imposed on the problem [12–14]. Each individual (also known as a chromosome) represents a solution to the problem. A chromosome consists of a set of parameters called genes, which are usually represented in a binary bit string. Chromosomes evolve through continuous iterations (known as generations). They are evaluated, and their fitness values are computed using an objective function during each generation. New chromosomes (known as offspring or children) are formed by applying crossover and mutation operations to generate the next generation. After a sequence of generations, the fittest chromosome is selected to represent the optimal or near-optimal solution to the problem [12].

In the literature, several studies have addressed the optimization of the task and human resource allocation in software project-scheduling problems by minimizing the project time. To our knowledge, none of those studies have addressed optimizing SDLC models. Moreover, no study has combined SDLC model selection and optimization. In this study, we focus on the selection of SDLC models and optimization of the completion time for the entire particular model by assigning the duration of each phase instead of allocating the tasks.

GAs have been successfully applied to different search and optimization problems. As mentioned, the main reasons for the success of GAs are their ease of use, flexibility, accuracy, and broad applicability. Therefore, we apply a GA to solve our problem [11].

3 Proposed SDLC Selection and Optimization Methodology

In our study, we propose a new methodology that solves the SDLC optimization problem using a GA. The methodology selects the best SDLC and optimizes the completion time of the selected model. This study aims to help project managers select the proper SDLC model for their projects and optimize the selected model. The SDLC model selection approach is based on a selection matrix. It consists of a set of selection criteria and information related to the project's nature given by the project managers. Our methodology optimizes the selected SDLC model by reducing the project completion time and assigning duration for each phase. The methodological framework has two main parts, as shown in Fig. 1. The first part is the selection process. The overall objective is the selection of the proper SDLC model; it contains the selection matrix module that is responsible for the decision of choosing the best model. The second part is the optimization process for the selected model; it contains the GA module. The overall objective of this part is to reduce the completion time of the project and assign duration for each phase in an optimal way.

The pseudocode of the SDLC selection and optimization methodology is given in Fig. 2. The selection of an SDLC process (Step 1) starts when the project manager/decision maker enters the required information related to the given project, such as the total number of employees (project team) involved in the project development. The project manager also determines the project characteristics on the basis of a set of

selection criteria. These criteria are used in the selection matrix to select an appropriate SDLC methodology among different SDLCs. Further details of the selection mechanism and matrix are provided in Subsection 3.1. The best SDLC model that suits the project is selected, and the number of selected SDLC phases and the number of repetitions for each phase are calculated. Then, the process of optimizing the selected SDLC via a GA is applied in Step 2. In this step, the selected SDLC is optimized by assigning durations for each SDLC phase with the objective of minimizing the project time. Before the optimization process is run, the initial solution (population) is constructed with an appropriate representation based on the selected SDLC. Once the initial solution is fully constructed, the optimization process is started. Additional details of the GA process are in Subsection 3.2. Eventually, the optimal completion time of the project and the optimal duration for each SDLC phase are obtained.

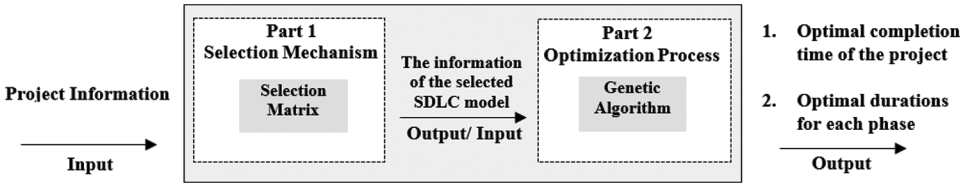

Figure 1: Proposed framework for our methodology

```
Input: project information (estimated number of employees, estimated
budget..etc..)
Begin
Step1 Selection  Mechanism
1.1 Selection criteria ← S;
1.2 SDLC a ← a[S];
    SDLC b ← b[S];
……
1.3 Set Project i information by project manager  ← i[S];
1.4 Evaluate i[S] by selection matrix;
1.5 Select best SDLC for project i;
Step 2 Optimization selected SDLC based on  GA
2.1 t ← 0;
2.2 Initialize P(t);
2.3 Evaluate P(t);
2.4 Repeat
2.5 crossover to create C(t) from P(t);
2.6 mutation to create C(t) from P(t);
2.7 Evaluate C(t);
2.8 Select P(t +1) from P(t) and C(t);
2.9 t ← t +1;
2.10 Until meet the termination condition
output: optimal completion time for selected SDCL and the assigned
duration for each phase)
end
```

Figure 2: Pseudocode of the SDLC selection and optimization methodology

3.1 SDLC Selection Mechanism

We introduce a new mechanism for the selection of the SDLC model. We constructed a selection matrix based on a set of selection criteria. We compared the most popular SDLC models based on 11 criteria. Each criterion has a predefined set of values, such as *clear/unclear* and *small/medium/large*. For instance, the requirements should be clear at the beginning of the software development process in waterfall projects.

We evaluated the most common models, namely, waterfall model, IID model, spiral model, agile model, RAD model, and software prototyping. Then, we set their values. Before the selection of any model, all models were evaluated by our methodology according to their fitness with respect to the project information given by the project manager. Subsequently, the ranking values for each model were obtained by computing the 11 values for every model. Finally, the model with the highest ranking value was selected to be the best model for the project development. The proposed selection matrix is provided in Tab. 1.

Table 1: Simulation parameters

Selection criterion	SDLC Models						
	Waterfal1	*Agile*	*Spiral*	*Prototyping*	*IID*	*RAD*	*V-model*
Clarity of requirements	C	UC	UC	UC	C	UC	C
Dealing with frequent changes	LW	H	H	H	H	LW	LW
Development time	SH	SH-M	SH	SH-M	SH	M-L	M-L
Project size	S	S-M	LG	M-LG	S	LG	S-M
Experience level of the project team	E	E	E	E	E	LE	LE
Risk management	LW	H	H	H	LW	H	LW
Complexity	SE	SE	CX	CX	SE	CX	CX
User involvement	NR	R	R	R	R	R	NR
Cost management	M	H	H	H	LW	H	M
Schedule visibility	NV	V	V	V	V	V	NV
Documentation	MR	LS	LS	MR	LS	MR	MR

Note: **C:** Clear, **UC:** Unclear, **LW:** Low, **M:** Medium, **H:** High, **SH:** Short, **L:** Long, **S:** Small, **LG:** Large, **E:** Experienced, **LE:** Less experienced, **SE:** Simple, **CX:** Complex, **NR:** Not required, **R:** Required, **NV:** Nonvisible, **V:** Visible, **MR:** More, **LS:** Less.

From a review of many studies, we identified the following criteria and their values that influence the choice of the SDLC model:

- **Clarity of requirements** *(clear/unclear)*: At the beginning of the software development process [2,3,5,15,16].
- **Dealing with frequent changes** *(low/high)*: During software development [2,5,15,16].
- **Development time** (*short/medium/long*): To accomplish the development process [2,15].
- **Project size** *(small/medium/large)* [2,16].
- **Experience level of the project team** (*less experienced/experienced*) [2,17].
- **Risk management** *(low/high)*: If the model accommodates risk management [2,17].
- **Project complexity** *(simple/complex)* [2,3,15,16].
- **Cost management** (*low/medium/high*): If the model accommodates cost management [3].
- **User involvement** *(required/not required)*: During the software development process [1,3,5,16].
- **Schedule visibility** *(visible/nonvisible)*: During the software development process.
- **Documentation** (*less/more*): Emphasis on documentation during software development [3].

The data provided by the selection process are the name of the best model for the project, number of phases, and number of iterations for each phase. These data were used as inputs in GA optimization.

3.2 SDLC Optimization Based on GA

3.2.1 Solution Representation

In addition to the selection of the best SDLC model for the project, we optimized the completion time of the project and assigned duration for each phase of the selected SDLC model.

Initially, we represented the problem solution. In GA, each individual of the population represents a possible solution to the problem. In the case of our problem, an individual is considered an instance of the SDLC model. The individual represented by string of genes is known as a *chromosome*. Each *gene* in the chromosome represents one phase of the SDLC model. The values of a gene represent three factors: (i) repetition factors, which include the number of iterations for each phase; (ii) the phase duration assigned by the algorithm; and (iii) the number of employees assigned for each phase in accordance with the total number of employees given by the project manager. The data provided by the SDLC selection process are used to construct the initial solution with appropriate representation. The chromosome length differs based on the number of phases of the selected SDLC model. Fig. 3 illustrates an example chromosome representation.

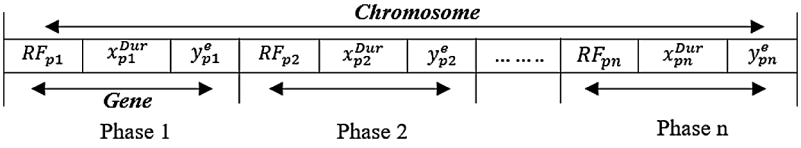

Figure 3: Example of chromosome representation

A GA searches for the optimal solution of a problem. It calculates the objective values and selects those individuals that obtain the best objective values (selection). The example in Fig. 4 exhibits how the crossover and mutation operations are performed in our algorithm. We designed multipoint crossover and chose crossover points in every gene in the chromosome. Those points are the phase duration portion in each gene. The crossover operator swaps the values of parents among themselves for every crossover point. The mutation operator randomly changes the value of the phase duration of a randomly selected gene of the offspring.

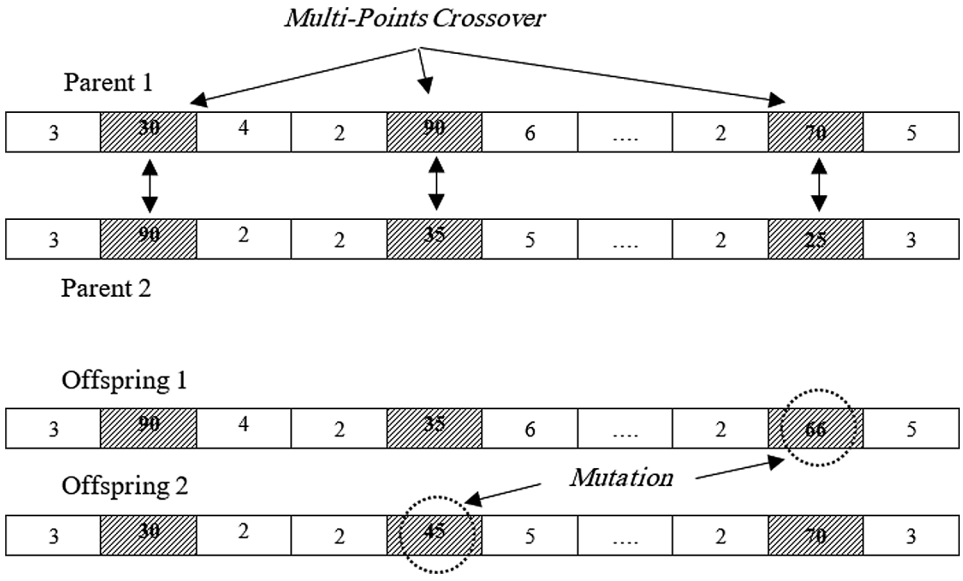

Figure 4: Example of crossover and mutation operations

3.2.2 Problem Formulation

We seek to reach the solution that minimizes the completion time of a given project. Therefore, our optimization problem can be formally described as

$$minf(x) = \sum_{i=1}^{P=N} \frac{x_{pi}^{Dur} \cdot RF_p i}{y_{pi}^{e}} \forall\, pi \in P \tag{1}$$

It is restricted by $1 \leq y \leq E$, where E is the total number of employees for a given project that the project manager must determine beforehand. The number of employees for each phase is assigned in a way that satisfies the constraint $1 \leq y \leq E$. The repetition factor (number of iterations for each phase) is constant for each phase in a given model. That is, the repetition factor in the individuals' representation remains constant. The notations indicated in Tab. 2 were used to define Eq. (1).

Table 2: Notations for problem formulation

Notations	Descriptions
$\boldsymbol{P}$	Number of phases in an SDLC model
$\boldsymbol{N}$	Variable number of phases in an SDLC model
pi	Phase i, phase 1, 2, 3, .., etc. differ in accordance with the selected model
e	Number of employees assigned for each phase
Dur	Duration assigned for each phase (phase completion time)
Rf	Repetition factor for each phase
$\boldsymbol{E}$	Total number of employees given by the project manager
$\boldsymbol{x_{pi}^{Dur}}$	Duration assigned for phase i, variable for each phase
$\boldsymbol{y_{pi}^{e}}$	Number of employees assigned for phase I, variable for each phase
$\boldsymbol{x}$	Variable duration
$\boldsymbol{y}$	Variable employee number
$\boldsymbol{RF_{pi}}$	Number of iterations for phase i

Thus, the problem solution consists of a combination of values that minimizes the completion time of the project. Those values represent the duration assigned by the algorithm to each phase of the selected model. The objective ***f (x)*** sums up the total duration values for all phases and results in the objective value. The individual with the lowest objective value will be selected by the algorithm to be the optimal solution.

4 Experimental Settings

We conducted a total of 150 experiments on the five most popular SDLC models (waterfall, RAD, V-model, spiral, and agile) using a GA. We divided the experiments among the five models, each representing an example of a particular project with a certain employee number. We repeated them 30 times. In this study, we analyzed how our algorithm can optimally minimize the completion time of a given project and assign duration for each phase. We implemented the GA in Java 8 using the NetBeans IDE 8.2 platform and performed 30 independent runs for each model on an Intel CORE i7, 8th Gen, 2.0 GHz, 8 GB of RAM PC running Windows 10. A population of 64 individuals and multipoint crossover and mutation were applied. The crossover probability was set to 1.0, and the mutation probability was set to 0.07. The stopping criterion was set to reach the maximum generation, which was 500. We set the values of the GA parameters based on a survey of studies that have used GAs to solve similar optimization problems [10].

We believe that these studies have provided good solutions to optimization problems. Tab. 3 illustrates the parameter settings that we set for the GA runs. Before running the GA, our methodology constructed the initial solution with an appropriate representation based on the selected SDLC model. Our GA optimization was implemented with a range of durations that were randomly assigned for the SDLC model phases. Each range of durations was set to reflect the real completion time of a given model. Accordingly, the duration of the phases was assigned randomly to its appropriate position in the genes. The number of employees for each phase was also assigned with a random distribution to its appropriate position in the genes according to the total number of employees given by the project manager. We set a range of total number of employees for every model. These ranges reflect the real sizes of project teams required to accomplish the project. When we conducted independent experiments for each model, we considered various employee ranges.

Table 3: Parameter settings of the GA

GA Parameters	Settings
Population Size	64
Length of individual chromosome	Differs based on the selected SDLC model
Stopping criteria	Max. generation
Max. generation	500
Selection type	Roulette Wheel Selection
Crossover rate	1.0
Crossover type	Multipoint crossover
Mutation rate	0.07
Number of independent runs for each model	30

5 Experimental Results and Discussion

Several experiments were conducted to obtain the optimal completion time for the selected SDLC models. Tab. 4 shows the results obtained in the experiments for the five models, where the first column denotes the experiment number, and the second to the sixth columns indicate the objective values (optimal completion times in days) for each model. The boxplot analysis of the results is depicted in Fig. 5, which exhibits the distribution of the experimental results. Our GA was originally implemented to handle different ranges of phase durations to reflect the real completion time. Accordingly, Tab. 4 presents the distribution of the results, that is, completion time of the project, which varies with the model type. The results may also be affected by several factors, such as the number of genes (number of phases of the SDLC model), the total number of employees and their distribution at each phase, and the iteration number.

Tab. 5 presents the minimum, maximum, and standard deviation of the obtained results for the five models, and Tabs. 6–10 show the minimum, maximum, and standard deviation of the obtained results in detail for each model. From the statistical results of the SDLC models presented in Tab. 5, the waterfall model had high *StDev*, which meant that significant differences existed among objective values. The agile and spiral models had medium *StDev*, and V-model and RAD model had low *StDev*. The reason for the high *StDev* in the waterfall model was the large range set to assign the durations of the phases of the waterfall model randomly. The *StDev* in V-model and RAD was low due to fewer ranges. Although the distribution of the objective values obtained in the experiments varied with the model type, these results were realistic and optimal completion times of the projects.

Table 4: Experimental results for the five SDLC models

Objective Values (Optimal completion times of the project in days)					
Exp. #	Waterfall	Agile	Spiral	V-model	RAD
1	834	450	198	279	157
2	855	420	231	279	151
3	902	420	153	275	153
4	842	390	171	279	151
5	791	420	162	282	151
6	841	420	171	278	154
7	723	360	171	277	156
8	851	420	153	281	151
9	784	450	168	278	152
10	839	450	186	281	155
11	847	420	177	284	151
12	842	420	144	289	156
13	894	390	162	281	155
14	781	450	162	282	153
15	781	390	162	276	151
16	842	390	208	283	152
17	843	360	162	276	155
18	844	450	162	279	155
19	847	390	162	285	152
20	787	360	153	279	150
21	780	390	162	281	156
22	846	420	178	279	154
23	783	390	162	283	152
24	776	390	153	281	154
25	837	420	177	278	152
26	723	392	168	285	153
27	781	390	153	284	153
28	1016	420	168	282	154
29	1019	360	177	284	153
30	840	420	153	284	155

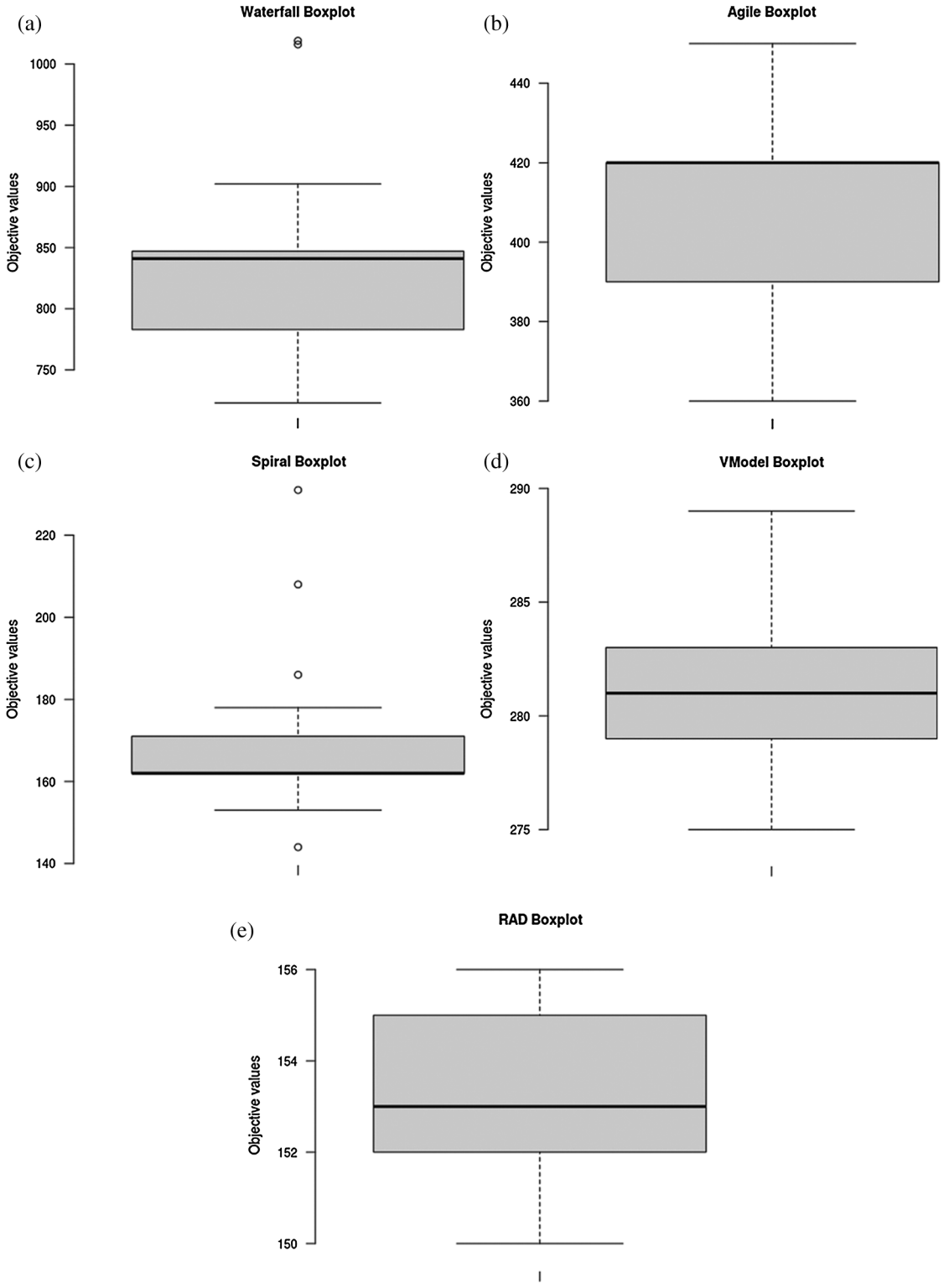

Figure 5: Boxplot analysis of the results. (a) Waterfall; (b) agile; (c) spiral; (d) V-model; (e) RAD

Table 5: Statistical results for the five SDLC models

Statistics	Waterfall	Agile	Spiral	V-model	RAD
Min	723	360	144	275	150
Max	1019	450	231	289	157
StDev	***66***	***28***	***18***	***3***	***2***

Table 6: Statistical results for the waterfall model

	Optimal time (objective values)		Requirement analysis	System design	Implementation	Testing	Deployment	Maintenance
	D	M						
Min	723	24	118	118	118	118	118	119
Max	1019	33	358	186	359	183	183	189
StDev	***66***	***2***	***49***	***25***	***47***	***30***	***27***	***26***

Note: D: Days M: Months

Table 7: Statistical results for the agile model

	Optimal time (objective values)		Planning	Requirement analysis	System design	Development	System testing	Deployment and maintenance
	D	M						
Min	360	12	60	60	60	60	60	60
Max	450	15	90	90	92	90	90	90
StDev	***28***	***1***	***11***	***12***	***14***	***14***	***14***	***13***

Table 8: Statistical results for the spiral model

	Optimal time (objective values)		Planning/requirement identification	System design	System building	Evaluation and risk analysis
	D	M				
Min	144	4	36	36	36	36
Max	231	7	61	90	90	91
StDev	***18***	***1***	***8***	***11***	***10***	***12***

Table 9: Statistical results for the V-model

	Optimal time (objective values)		Requirement and analysis	User acceptance testing	System design	System testing	Architectural design	Integration testing	Module design	Unit testing	Implementation
	D	M									
Min	275	9	30	30	30	30	30	30	30	30	30
Max	289	9	33	37	36	33	35	35	34	38	37
StDev	***3***	***0***	***1***	***2***	***1***	***1***	***1***	***1.2***	***1.3***	***2***	***2***

Table 10: Statistical results for the RAD model

	Optimal time (objective values)		Business modeling	Data modeling	Process modeling	Application generation	Testing and turnover
	D	M					
Min	150	5	30	30	30	30	30
Max	157	5	34	34	36	35	32
StDev	***2***	***0***	***1***	***1***	***1***	***1***	***1***

From the results of the waterfall model experiments presented in Tab. 6, the *Min* value was 723, *Max* value was 1019, and *StDev* was 66. Analysis of the waterfall boxplot results shown in Fig. 5a indicates that the distribution of the objective values was realistic and optimal as completion time of the waterfall model projects. The significant differences among objective values (*StDev*) obtained in the experiments were normal due to the random distribution of the significant duration range suitable for the completion time of the waterfall model projects. Indeed, objective values may also be affected by several factors, such as the number of genes (number of waterfall phases) and the employee distribution for each phase. With respect to the agile model shown in Tab. 7 and Fig. 5b, the *Min*, *Max*, and *StDev* were 360, 450, and 28, respectively. Significant differences existed among the objective values (*StDev*) obtained in the experiments. These results were expected and realistic for the agile model projects. This is because the significant duration range of the project completion time for the agile model affected the random distribution of objective values. In addition, the agile model was affected by several factors, such as the number of genes (number of the agile phases) and the number of model iterations. The results may further be affected by the employee distribution for each phase. The spiral model results presented in Tab. 8 and the visual results shown in Fig. 5c indicated that the distribution of objective values was realistic and optimal for the project completion time of the spiral model. The differences among objective values obtained in the experiments were normal and considerably high due to the random distribution of few duration ranges suitable for the project completion time of the spiral model. The results may also be affected by several factors, such as the number of genes (number of spiral phases), the number of iterations, and the employee distribution for each phase. The results of the V-model experiments presented in Tab. 9 and the boxplot results shown in Fig. 5d demonstrate that no significant differences existed among the objective values obtained in the experiments. The results seem stable, realistic, and optimal for V-model. These results are normal and expected due to the random distribution of the very few duration ranges suitable for the project completion time of the V-model. The RAD model experimental results in Tab. 10 and Fig. 5e indicate no significant differences among the objective values obtained in the experiments. The results are stable, realistic, and optimal for the project completion time of the RAD model. These results were expected due to the random distribution of very few duration ranges suitable for the project completion time of the RAD model. The experiments in this study were limited only to the five SDLC models, which are waterfall, agile, spiral, V-model, and RAD. Other models, such as IID and prototyping, were excluded because they rely on an iterative factor that needs an intensive computation.

6 Conclusion and Future Work

In this study, a new methodology for SDLC model selection and optimization is proposed. No approach or study exists in the literature that addresses the optimization of SDLC models. This gap motivated us to conduct this study. We proposed a methodology that would provide a good solution for project managers in software project organizations. The proposed methodology could help project managers in the selection

of a proper SDLC model and optimize completion time of the project in the selected model. The selection of a proper SDLC model depends on many criteria related to the nature of the software project, project team, experience level of the developers, and risk and cost management. We identified a set of selection criteria on the basis of many studies and introduced a selection matrix into SDLC model selection. Moreover, we applied a GA to optimize the selected model with the objective of minimizing the completion time of the project. We implemented the GA in a way that represented our solution. We conducted a total of 150 experiments on the 5 most popular SDLC models (waterfall, RAD, V-model, spiral, and agile). These experiments demonstrated how our algorithm can optimally minimize the completion time of a given project and assign duration for each phase. The obtained results of the experiments showed that the objective values for the five models were realistic and optimal as completion times of projects.

In the future, other SDLC models can be included in the SDLC selection matrix. The application of other search-based techniques to the SDLC optimization problem needs further investigation. In addition, the performance of the results could be improved by combining GAs with other optimization techniques, such as HC and SA.

Acknowledgement: The authors would like to thank the Deanship of Scientific Research at King Saud University for funding and supporting this research through the initiative of DSR Graduate Students' Research Support.

Funding Statement: The authors received funding from the Deanship of Scientific Research at King Saud University for this study. F. Albalawi. https://dsrs.ksu.edu.sa/ar/node/2228.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. K. Sharma, "A study of SDLC to develop well engineered software," *Int. Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 520–523, 2017.

[2] V. Öztürk, "Selection of appropriate software development life cycle using fuzzy logic," *Journal of Intelligent & Fuzzy Systems*, vol. 25, no. 3, pp. 797–810, 2013.

[3] M. Hicdurmaz, "A fuzzy multi criteria decision making approach to software life cycle model selection," in *Proc. of the Euromicro Conf. on Software Engineering and Advanced Applications (SEAA)*, Cesme, Izmir, pp. 384, 2012.

[4] K. Ali, "A study of software development life cycle process models," *Int. Journal of Advanced Research in Computer Science*, vol. 8, no. 1, pp. 15–23, 2017.

[5] M. A. Ben-zahia and I. jaluta, "Criteria for selecting software development models," in *Proc. of the Global Summit on Computer & Information Technology (GSCIT)*, Sousse, pp. 1–6, 2014.

[6] H. M. E. Abdelsalam and H. P. Bao, "A simulation-based optimization framework for product development cycle time reduction," *IEEE Transactions on Engineering Management, Engineering Management*, vol. 53, no. 1, pp. 69–85, 2006.

[7] N. Bibi, Z. Anwar and A. Ahsan, "Comparison of search-based software engineering algorithms for resource allocation optimization," *Journal of Intelligent Systems*, vol. 25, no. 4, pp. 629–642, 2016.

[8] M. Harman, S. A. Mansouri and Z. Yuanyuan, "Search-based software engineering: Trends, techniques and applications," *ACM Computing Surveys*, vol. 45, no. 1, pp. 11:1–11:61, 2012.

[9] A. V. Rezende, "Software project scheduling problem in the context of search-based software engineering: A systematic review," *Journal of Systems and Software*, vol. 155, no. 6, pp. 43–56, 2019.

[10] M. Á. Vega-Velázquez, A. García-Nájera and H. Cervantes, "A survey on the software project scheduling problem," *Int. Journal of Production Economics*, vol. 202, no. 11, pp. 145–161, 2018.

[11] E. Alba and J. F. Chicano, "Software project management with GAs," *Information Sciences*, vol. 177, no. 11, pp. 2380–2401, 2007.

[12] A. Kumar, *Network design using genetic algorithm*. LAP LAMBERT Academic Publishing, Saarbrücken, Germany, 2016.

[13] M. A. Mohammed, M. K. Abd Ghani, N. Arunkumar, R. I. Hamed, M. K. Abdullah *et al.,* "A real time computer aided object detection of nasopharyngeal carcinoma using genetic algorithm and artificial neural network based on Haar feature fear," *Future Generation Computer Systems*, vol. 89, no. 4, pp. 539–547, 2018.

[14] M. A. Mohammed, M. K. Abd Ghani, R. I. Hamed, S. A. Mostafa, M. S. Ahmad *et al.,* "Solving vehicle routing problem by using improved genetic algorithm for optimal solution," *Journal of Computational Science*, vol. 21, no. 10, pp. 255–262, 2017.

[15] M. A. Khan, A. Parveen and M. Sadiq, "A method for the selection of software development life cycle models using analytic hierarchy process," in *Proc. of the Int. Conf. on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Ghaziabad, pp. 534, 2014.

[16] P. M. Khan and M. S. Beg, "Extended decision support matrix for selection of SDLC models on traditional and agile software development projects," in *Proc. of the Third Int. Conf. on Advanced Computing and Communication Technologies (ACCT)*, Rohtak, pp. 8, 2013.

[17] N. Budhija and S. P. Ahuja, "Study of software process model selection," *Int. Journal of Advanced Research in Computer Science*, vol. 2, no. 6, pp. 279–28220, 2011.