Journal of Engineering Technology
Volume 8, SI (Recent Trends in System Engineering), April 2019, PP. 108-128
RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

Methods for the Selection of Software Requirements: A Literature Review

Mohd. Sadiq[1], Chaudhary Wali Mohammad[2], and Shadab Khan[2]

[1]Computer Engineering Section, UPFET, Jamia Millia Islamia (A Central University), New Delhi, India
[2]Computer Science and Technology Research Group, Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia (A Central University), India

Abstract: Software requirements selection (SRS) from the pool of requirements for the different releases of the software is a general problem faced by many software industries. Different methods have been proposed to solve the SRS problem like integer linear programming, metaheuristic algorithms, approximate backbone-based multilevel algorithms, analytic hierarchy process, etc. The objective of this paper is to review the literature from the SRS research area. Therefore, related articles appearing in the International Journals and Conferences from January 1996- December 2017 are gathered and analyzed so that the following five research questions can be answered: (i) which methods were frequently applied for SRS? (ii) which evaluating criteria have received more attention in the area of SRS? (iii) how many methods have applied fuzzy based approaches for SRS? (iv) is there any support in the SRS methods to deal with the requirements uncertainty? (v) Is there any limitation of the existing approaches? Based on the insufficiency, if any, some improvements and possible future work in the area of SRS are recommended.

Keywords: Software Requirements selection, Next release problem, Integer linear programming, Analytic hierarchy process, Metaheuristic algorithms, Quality function deployment, Fuzzy based methods.

1. Introduction

Software requirements selection (SRS) from the pool of requirements is a difficult task because large number of requirements needs to be processed during different releases of software [1]. In "*search based software engineering*" (SBSE), SRS problem is also referred to as "*Next Released Problem*" (NRP). NRP is classified into two parts, i.e., (i) single objective NRP and (ii) multi-objective NRP. Single objective NRP was formulated by Bagnall *et al*. [2] in 2001; and its objective is "*to find the ideal set of requirements that balance the customer requests within resource constraints*". NRP is a classical instance of "*0/1 knapsack problem*". It is an NP hard problem because it involves several contrary objectives that have to be addressed by the engineers. SBSE methods have proven to be essential in requirements optimization [3].

Multi-objective NRP was formulated by Zhang *et al*. [3] in 2007, whose objective was to select the software requirements "*by minimizing the cost and maximizing the customer satisfaction*". In Figure 1, we classify the methods used for the SRS based on the following criteria, i.e., (i) when large number of criteria are used in SRS (ii) when few objective criteria are used in SRS. In literature, we identify that two methods are generally used for SRS when large numbers of criteria are given, i.e., "*Analytic Hierarchy Process"* (AHP) [4] and "*Quality Function Deployment*" (QFD) [5]. In SBSE, SRS problem can be solved by the following techniques: (i) "*Integer Linear Programming*" (ILP), (ii) Metaheuristics algorithms, which include swarm intelligence and evolutionary algorithms [22], etc., and (iii) approximate backbone-based multilevel algorithms [20].

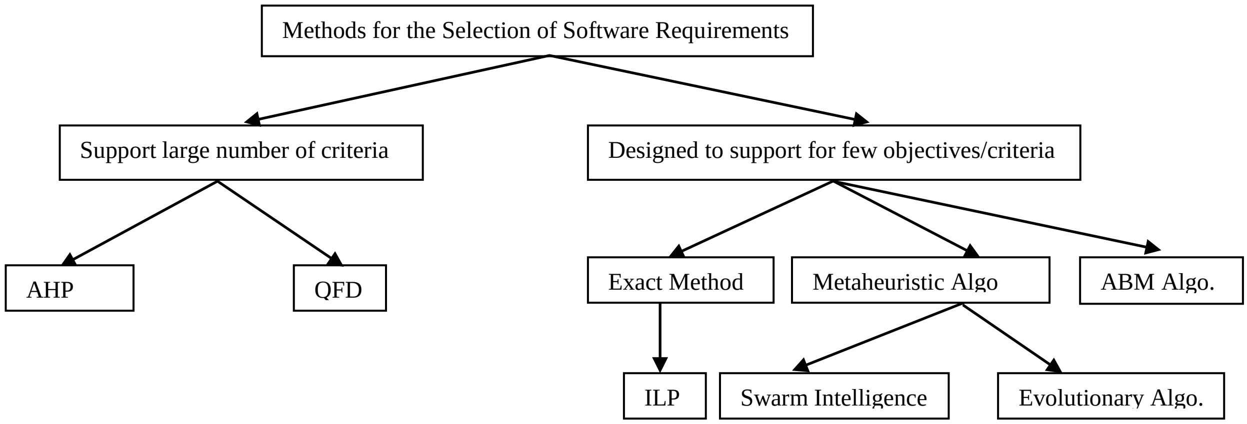

Figure.1. Classification of the methods used for solution of SRS problem

Requirements engineering (RE) is a process which is used to identify the different types of the software requirements with the help of the different requirements elicitation techniques. Once we have identified the software requirements then the next step is to select the software requirements that would be implemented during different releases of the software [6]. SRS is an important activity of requirements management which keeps track of all the requirements changes. During SRS, security, effort and cost are considered as important criteria for the selection of SR [7].

There are few articles which have reviewed the literature regarding the "*software requirements selection and prioritization*" (SRSP). For example, Pitangueira et al. [8] identified 39 articles for SRSP using SBSE. In a similar study, Pitangueira et al. [9] identified 30 papers for the systematic review for the SRSP using SBSE. These studies were limited to only SBSE methods. In SBSE area, only few objectives are employed for the solution of SRS problem [8]. In real life application, large number of criteria may be involved for the solution of SRS problem. In literature, we identify that existing studies do not support a literature review in the area of SRS method when the few as well as large number of criteria are used for the solution of SRS problem. Therefore, to address this issue, we select and review the published literature from January 1996 to December 2017; and present a comprehensive overview of existing methods used in SRS.

Based on 68 selected studies retrieved from ScienceDirect, Springer, IEEE Explore, and Google Scholar, following research questions (RQs) are examined:

- RQ-1: Which methods were frequently applied for SRS?
- RQ-2: Which evaluating criteria have received more attention in the area of SRS?
- RQ-3: How many methods have applied fuzzy based approaches?
- RQ-4: Is there any support in the SRS methods to deal with the requirements uncertainty?
- RQ-5: Is there any limitation of the existing approaches?

The main contributions of the paper are given as below:

- Identification of the high quality studies in the field of SRS
- Classification of the SRS methods based on number of criteria used during the SRS process
- Suggestions for future research direction in the area of SRS

The remainder of the article is structured as follows: Section 2 describes the individual methods used in the selection of software requirements. In section 3, we discuss the groups of methods applied separately for SRS. Integrated methods for SRS are given in section 4. Section 5 presents the detailed description of the findings based on the research questions. Finally, section 6 concludes the paper and suggests future work in the area of SRS.

2. Individual Methods

In this section, we present individual methods which are used to select the software requirements. We have identified 31 studies out of 68 (45.59%) in which individual methods have been used for the selection of software requirements.

2.1. Integer Linear Programming

We have identified nine studies out of 68 (13.24%) in which "*integer linear programming*" (ILP) was used to solve the SRS problem. The detailed description of the dataset, application, and evaluating criteria used in the method are summarized in Table 1.

Jung [10] developed a method for SRS, which is an extension of the work by Karlsson and Ryan [4], by considering the variant of "*0/1 knapsack problem*". Karlsson and Ryan [4] used AHP method for the requirements selection. Therefore, to minimize the complexity of AHP method, Jung [10] apply the integer programming for the SRS.

Table 1: Integer Linear Programming-Applications/Dataset and Evaluating criteria

Paper ID	Authors	Applications/ Dataset	Evaluating Criteria/ Objectives
ILP-1	Jung [10]	RAN Project and PMR Project	Importance of requirements and Cost
ILP -2	Van den Akker *et al.* [11]	On real life dataset with 9 requirements and 3 teams, 24 requirements and 17 teams and 99 requirements and 17 teams	Estimated revenue per requirements and Resources
ILP -3	Van den Akker *et al.* [12]	On real life dataset with 9 requirements and 3 teams, 24 requirements and 17 teams and 99 requirements and 17 teams	Estimated revenue per requirements and Resources
ILP -4	Li *et al.*[14]	Two data sets are used: (i) Small dataset: 9 requirements with 3 teams and release duration 60 days (ii) Master dataset : 99 requirements and 17 teams and release duration 30 days	Cost, Development time and Revenue
ILP -5	Van den Akker *et al.* [13]	Real life data	Estimated revenue per requirements and Resources
ILP -6	Li *et al.* [76]	Two data sets are used: (i) Small dataset: 9 requirements with 3 teams and release duration 60 days (ii) Master dataset : 99 requirements and 17 teams and release duration 30 days	Project duration, Revenues, and On Time

			delivery
ILP -7	Sureka [15]	Tested on synthetic dataset: (i) 50 requirements with additive value for all combination of requirements (ii) 50 requirements with non-additive values for requirements packages	Cost and importance of requirements
ILP -8	Veerapen *et al.* [16]	Motorola and University College London dataset	Cost and Profit
ILP-9	Mougouei [17]	Simulation based study	Budget

Van den Akker *et al.* [11] applied ILP for the determination of the set of the "*software requirements for the next release of the software product*". In this method, candidate requirements, estimated revenue per requirements, and the available resources are used as input; and finally a set of selected requirements are generated as an output. In another study, Van den Akker *et al.* [12] proposed a method for the next release of the software product using ILP. In their work, a tool was developed by using ILP to help the product managers as well as project managers during the release planning (RP). This tool firstly accept the following as input, i.e., "*the list of candidate requirements*", "*estimated revenue*", and "*required team resources per requirement*"; and then the following information is used to get the set of selected requirements, i.e., "*team composition*", "*permitting of team transfers*", "*extension of deadlines*", and "*hiring external resources*".

Van den Akker *et al.* [13] developed a tool using ILP to support the product and project managers during software release planning. This tool generates the "*optimal set of requirements with maximum projected revenue against available resources*".

Li *et al.* [14] investigated two ILP models by integrating the requirements scheduling into software RP. The objective of the first model was to minimize the project span and to satisfy the precedence constraints by developing the requirements for the next release of software. Second model was used to combine the requirements selection and scheduling. This model not only maximizes the revenues but also computes an on time delivery project schedule simultaneously. Li *et al.* [76] integrate the scheduling with SRS by considering the "dynamic *adaptation for over estimation or under estimation of revenues or processing time*".

Sureka [15] extended the "*mathematical formulation of the NRP under non-additive customer valuations*". An ILP formulation of NRP was incorporated with the "*positive and negative synergy between requirements across all the combinations of the requirements*". An experimental work was carried out by considering the following algorithms, i.e., *Non-dominated Sorting Genetic Algorithm*" (NSGA) -II, *Multi-objective Evolutionary Approach*" (MOEA), i.e., εMOEA, MOEA/D, etc. ILP is convenient method for solving both type of NRP, i.e., single objective and bi-objective NRP. Veerapen *et al.* [16] revisited the ILP for single objective NRP and integrate an "*epsilon-constraint method*" with ILP to address the bi-objective problem. NSGA-II was used for the comparison with the proposed method. As a result authors found that exact algorithms can be used for "*single objective instances and small bi-objective instances*". Mougouei [17] proposed a method for SRS by considering the dependency among the requirements using ILP. The dependency among the requirements was represented by the graph based dependency model.

2.2. Metaheuristic Algorithms

Among 68 articles, fourteen papers (20.59%) apply metaheuristic algorithms for the selection of software requirements. The detailed description of the dataset, application, and evaluating criteria used in the methods are summarized in Table 2.

2.2.1 Swarm Intelligence

Sagrado and Aguila [18] proposed a method to show that "*ant colony optimization*" (ACO) algorithm can be used to solve the software requirements selection problem. They suggested that ACO system should be compared with Greedy and Simulated Annealing algorithms. Sagrado *et al.* [19] used ACO algorithm for SRS problem. They have evaluated the proposed method with other algorithms, i.e., "*Simulated Annealing*" (SA) and "*Genetic Algorithm*" (GA). A case study was used to compare the ACO, SA, and GA.

Jiang *et al.* [20] employed "*multiple artificial ants*" to develop new solutions for the SRS problem. In their algorithm, both "*pheromone trails and neighboring information*" were used to determine the choice of every ant. First found hill climbing operator was incorporated into hybrid ACO (HACO). It was compared with "*greedy randomized adaptive search procedure*" (GRASP) and SA. As a result authors found that HACO perform better than GRASP and SA with regard to quality and running time.

Souza *et al.* [21] proposed a method to solve the SRS problem in the existence of dependent requirements using ACO. Proposed method was evaluated over seventy two synthetic datasets. As a result, authors found that ACO algorithm generates more accurate results then SA and GA. Chaves-Gonzalez *et al.* [22] formulated the SRS problem as multi-objective optimization problem and proposed a software requirements optimization method using "*artificial bee colony*" (ABC) to achieve the following objectives, i.e., cost and satisfaction; and three constraints. In the same year, Chaves-Gonzalez *et al.* [23] employed the "*Teaching Learning Based Optimization* (TLBO)" algorithm for the SRS with two objectives: "*the total software development cost*" and "*overall customer satisfaction*"; and three interaction constraints. The proposed method was applied on real data provided by experts. Ranjith and Marimuthu [24] apply the Multi-objective *"Teacher-Learning-Artificial-Bee Colony*" (TLABC) optimization for the selection of software requirements.

Sagrado *et al.* [25] applied ACO algorithm for SRS. The performance achieved by the ACO was compared with other methods, i.e., GRASP and "*Non-dominated Sorting Genetic Algorithm*" (NSGA), by means of computational experiments. Ferreira *et al.* [26] formulated the single objective interactive NRP, i.e., iNRP. An ACO based solution was proposed and evaluated, which can generate solutions by considering the "*user supplied subjective factors and metrics related to the selection of software requirements*".

2.2.2. Evolutionary Algorithms

Ruhe and Greer [27] presented a GA based method called the evolutionary planning of incremental software development under risk and resources constraints, i.e., EVOLVE+ method. The same research group used Genetic Algorithm to propose a method called EVOLVE to allocate the requirements in incremental releases [28]. EVOLVE method was developed to do the following: (i) "*to allocate the optimal requirements in different increments, (ii) a means of assessing and optimizing the degree to which the ordering conflicts with stakeholder priorities within technical precedence constraints; (iii) a means of balancing required and available resources for all increments*". Araujo and Paixao [29] discussed the need of the human opinion and its characteristics in the search space. Therefore, machine learning technique was applied to model the user in an "*Interactive Genetic Algorithm*" (IGA).

Silva *et al.* [30] used "*Multi-Objective Genetic Algorithms (MOGA)*" for SRS problem. To generate the initial population of MOGA "*Path Relinking based method*" was used. The performance of the method was evaluated with the help of MoCell and NSGA-II. Chaves-Gonzalez and Perez- Toledano [31] applied multi-objective version of the "*Differential Evolution*" (DE) for SRS. They have formulated the SRS as a "*multi-objective optimization problem*" with two objectives, i.e., cost and customer satisfaction; and with three interaction constraints.

Table 2: Metaheuristic Algorithms- Applications/Dataset and Evaluating criteria

Paper ID	Authors	Type of algorithms used					Applications/ Dataset	Evaluating Criteria/ Objectives
		ACO	ABC	GA	TLBA/ TLABC	DE		
MA-1	Sagrado and Aguila [18]	√					Not Validated	No criteria
MA-2	Sagrado *et al.* [19]	√					Test Problem with 20 requirements and 5 customers	Cost and Effort
MA-3	Jiang *et al.* [20]	√					Five randomly generated problems	Cost and Profit
MA-4	Souza *et al.* [21]	√					72 synthetic data set	Cost and Profit
MA-5	Chaves Gonzalez *et al.* [22]		√				Used real instances	Cost and customer satisfaction
MA-6	Chaves Gonzalez *et al.* [23]				√		Used real instances	Cost and customer satisfaction
MA-7	Sagrado *et al.* [77]	√					Experimental work was carried out on 2 instances	Cost and customer satisfaction
MA-8	Ferreira *et al.* [26]	√					Dataset based on Microsoft word, ReleasePlanner™ Software, and random dataset having 50, 25, and 100 requirements, respectively.	Cost, Value, and Precedence
MA-9	Ruhe and Greer [27]			√			Randomly generated data	Effort, Risk, and Resource Constraints
MA-10	Greer and Ruhe [28]			√			Sample project with 20 requirements	Effort, Benefit, and Penalty
MA-11	Araujo and Paixao [29]			√			Tested on randomly generated data	Cost and Budget
MA-12	Silva *et al.* [30]			√			Tested on artificial data	Cost and Profit
MA-13	Chaves-Gonzalez and Perez-Toledano [31]					√	Tested on 20 and 100 requirements	Cost and Customer satisfaction
MA-14	Ranjith and Marimuthu [24]				√		Tested on two real dataset: (i) 20 necessities, 5 consumers with 10 necessities; (ii) 100 necessities, 5 clients, and 44 necessities.	Minimum cost, Maximum client satisfaction, Minimum time consumption

			and maximum reliability

2.3 Approximate Backbone-based Multilevel Algorithms

We have identified two papers (2.94%) based on backbone-based multilevel algorithms. The detailed description of the dataset, application, and evaluating criteria used in the method are summarized in Table 3.

As discussed in the literature that GA, ACO, simulated annealing, etc., can work effectively on the small scale NRP. Therefore, in order to deal with large set of requirements, Jiang *et al.* [32] proposed an "*approximate backbone based multilevel algorithm (ABMA)*" for the solution of SRS problem. ABMA explores the search space by "*multilevel reductions and refinements*". It was the first study which uses backbone in the area of requirements engineering. In ABMA, greedy climbing search operator is used to generate the local optimal solution instead of SA algorithm proposed by Lundy and Mees (LMSA) [33]. LMSA can work well on small scale instances of NRP.

In another study, Xuan *et al.* [34] applied "*backbone-based multilevel algorithm"* (BMA) for SRS from the large set of data. They have used two kinds of backbones for SRS problem, i.e., "*approximate backbone and soft backbone*". Using backbone, a large scale problem can be broken into small ones and can refine the solution to the original one.

Table 3: Backbone-based Multilevel (BM) Algorithms: Applications/ Dataset and Evaluating Criteria

Paper ID	**Authors**	**Applications/ Dataset**	**Evaluating Criteria/ Objectives**
BM-1	Jiang *et al.* [32]	Communication Company	Cost and Profit
BM-2	Xuan [34]	Evaluated on 39 NRP instances	Cost and Profit

2.4 Analytic Hierarchy Process

There are two papers (2.94%) using AHP in SRS. The detailed description of the dataset, application, and evaluating criteria used in the method are summarized in Table 4.

Karlsson and Ryan [4] proposed a method to determine the set of requirements for the implementation when a project has limited resources. Authors have used AHP to identify the requirements contribution. Requirements are pair-wise compared on the basis of the importance and cost of the requirements. Ruhe and Eberlein [35] proposed a method "*for trade-off analysis for the selection of software requirements*". In this method AHP was used to identify the preferences of stakeholders related to the various classes of requirements.

Table 4: Analytic Hierarchy Process (AHP)-applications/dataset and evaluating criteria

Paper ID	**Authors**	**Applications/Dataset**	**Evaluating Criteria/ Objectives**
AHP-1	Karlsson and Ryan [4]	Performance Management Traffic Recording System	Importance of requirements and Cost
AHP-2	Ruhe and Eberlein [35]	Text Processing Software System	Effort, Time, and quality

2.5 SRS with Multiple- stakeholders

There are three studies (4.41%) in which multiple stakeholders participated during SRS process. Their applications and evaluating criteria is given in Table 5. In order to deal the vagueness and impreciseness in the goal oriented requirements elicitation process for the selection of goals or requirements, Sadiq and Jain [36] proposed a fuzzy based approach. The first step of the proposed method includes the identification of the stakeholders who will participate during SRS. Proposed method was applied to select the requirements of Institute examination System. A method for the prioritization of the stakeholders was developed by Sadiq [37] on the basis of the importance of the software requirements.

In order to incorporate different stakeholders, Pitangueira [38] developed an interactive method for SRS. This method supports the search towards the selection of requirements according to the need of the stakeholders. Zhang *et al.* [39] discuss the need of different stakeholders and proposed an optimization based approach using two multi-objective evolutionary optimization algorithms. It was the first study which discusses the internal tensioning among different types of stakeholders.

Table 5: SRS with multiple stakeholders (MS) -applications and evaluating criteria

Paper ID	Authors	Application/Dataset	Evaluating Criteria/ Objectives
MS-1	Sadiq and Jain [36]	Institute examination system	Cost, Effort, and Risk
MS-2	Pitangueira [38]	Not validated	Cost, importance of requirements, and risk
MS-3	Zhang et al. [39]	Real world and synthetic dataset	Budget

2.6 Quality Function Deployment

There is only one study (1.47%) in which quality function deployment is used to select the requirements of software. Their applications, dataset, and evaluating criteria are given in Table 6. Sen and Baracli [5] proposed a fuzzy "*quality function deployment"* (QFD) methodology to select the software requirements based on non-functional criteria, i.e., quality, technology factors, and socioeconomic factors. A hierarchical structure of non-functional requirements was used to select the software's requirements.

Table 6: Quality Function Deployment-applications/dataset and evaluating criteria

Paper ID	Authors	Applications/Dataset	Evaluating Criteria/ Objectives
QFD-1	Sen and Baracli [5]	Turkey's electronic industry.	Quality, Technology, and Socioeconomic factors

2.7 Attributed Goal Oriented Requirements Analysis Method (AGORA)

There is only one study (1.47%) in which AGORA method has been used to select the requirements of software. Their application, dataset, and evaluating criteria are given in Table 7. Kaiya et al. [40] proposed an "*attributed goal oriented requirements analysis*" method. In this method, the goals/requirements are selected from the alternatives on the basis of the contribution and preference matrices.

Table 7: AGORA-applications/dataset and evaluating criteria

Paper ID	Authors	Applications/Dataset	Evaluating Criteria/ Objectives
AGORA-1	Kaiya *et al.* [40]	Web account system	Quality

3. Groups of Methods Applied Separately

We have identified eight studies (11.76%) in which different group of methods have been used for the selection of software requirements. Their application, dataset, and evaluating criteria are given in Table 8. Selection of software requirements using SBSE was first formulated by Bagnall *et al.* [2] and is known as Next Release Problem. This problem was solved by the following group of methods, i.e., ILP, Greedy algorithms, and local search techniques, i.e., Hill Climbing and SA.
Baker *et al.* [41] proposed an automated method to address the problem of "*determining the next set of releases in the course of software evolution*" by using the Greedy and SA algorithm. Proposed method was evaluated in large telecommunication organization. The first multi-objective formulation of the NRP was proposed by Zhang *et al.* [3]. They examined the four search techniques, i.e., (i) "*random search*", (ii) NSGA-II, (iii) "*a Pareto GA*, and (iv) "*a single –objective GA for the solution of MONRP*". On the basis of the empirical study, it was found that NSGA-II is fruitful for solving the MONRP.

Table 8: Group of methods applied separately -applications and evaluating criteria

Paper ID	Authors	Group of methods	Applications/ Experimentation/ Dataset	Evaluating Criteria/ Objectives
G-1	Bagnall *et al.* [2]	ILP, Greedy algorithms, Hill Climbing and, Simulated annealing	Five randomly generated problems	Profit and Cost
G-2	Baker *et al.* [41]	Greedy and Simulated Annealing algorithm	Mobile Telecommunication Device	Cost, Development time and Revenue
G-3	Zhang *et al.* [3]	NSGA-II, Pareto GA, and single objective GA	Consider three datasets, i.e., 40 requirements and 15 customers, 80 requirements and 50 customers, and 140 requirements and 100 customers.	Cost and total satisfaction of customers.
G-4	Durillo *et al.* [42]	NSGA-II, MoCell, and random search	2 empirical studies (ES): ES-1 has number of customers ranging from 15-100 and the number of requirements from 40-140. ES-2 has three instances, i.e., 20 requirements and 100 customers, 25 requirements and 100 customers, and 2 customers and 200 requirements.	Cost and total satisfaction of customers.
G-5	Durillo *et al.* [43]	NSGA-II, MoCell, and PAES	Motorola dataset having 35 requirements	Cost and Importance of requirements
G-6	Cai *et al.* [44]	NSGA-II, SPEA2, and IWO/MO	Random dataset and Motorola dataset	Cost and Importance of requirements

G-7	Mausa *et al.* [45]	Hill Climbing and Simulated Annealing algorithm	Tested on realistic datasets	Cost and Profit
G-8	Botelho *et al.* [46]	Ant Colony Optimization and Particle Swarm Optimization	Evaluated by using three group of large instances of NRP as discussed in Bagnall *et al.* (2001)	Cost and Profit

Durillo *et al.* [43] studied the NRP under multi-objective environment. They used NSGA-II, MoCell, and random search to evaluate the performance of the multi-objective NRP. Spread and hypervolume quality indicators were used to compare the results of the experimental work. After experimentation, authors found that MoCell presents better results with more than 20 requirements under spread quality parameter. Regarding the hypervolume quality indicator, both NSGA-II and MoCell have shown a similar performance with 20-40 requirements. By increasing the requirements it was found that, "*NSGA-II provides better results than MoCell*". Under both the quality indicators, "*random search obtained worst results in all instances*". In a similar study, Durillo *et al.* [42] applied 2 genetic algorithms, i.e., NSGA-II and MoCell, and one evolutionary strategy for solving the MONRP, i.e., PAES. Highest numbers of optimal solutions were obtained in case of NSGA-II. MoCell provides wide range of different solutions and PAES is the fastest technique but it provides the least accurate results.

Cai *et al.* [44] proposed a requirements dependency based multi-objective method for SRS. In their method, "*Multi-objective Evolutionary Approach*" (MOEA) i.e., NSGA-II, "*The Strength Pareto Evolution Algorithm-2*" (SPEA-2), and IWO/MO were used to provide the feasible solution. In their work, authors have also presented the improved version of the "*multi-objective invasive weed optimization (IWO)*" algorithm; and compared the results with existing multi-objective approaches. In their work, authors have used both "*synthetic and real world data*".

Mousa *et al.* [45] compared the performance of different SRS method, i.e., hill climbing and SA. They investigated the differences between 4 variations of hill climbing, i.e., steepest ascent, first found-ordered, first found-random, and sampling; and 2 variations of SA, i.e., (i) "*ordered and random*" and (ii) the "*random search algorithm*". Authors found that SA algorithm outperforms the hill climbing algorithms in solving the NRP.

Botelho *et al.* [46] addresses the scalability issue of the NRP and investigated two ways for solving the NRP by using the ACO and "*Particle Swarm Optimization*" (PSO) algorithm. After experimental work, authors found that PSO algorithm achieves superior results than ACO algorithm; and they also found that PSO and BMA have very similar results. In some instance, PSO algorithm has better performance than BMA.

4. Integrated Methods

We have identified twenty six studies (38.24%) in which different methods have been integrated to deal the problem of SRS. Their application, dataset, and evaluating criteria are given in Table 9.

Ruhe and Ngo-The [47] proposed a hybrid approach called EVOLVE* using evolutionary computing and multi-criteria decision making method. It is then combined by involving the human intelligence. There are three phases in the proposed method, i.e., modeling, exploration, and consolidation. Proposed method was demonstrated with the help of case study. In a similar study, Ruhe and Saliu [48] presented a hybrid approach of human intelligence and computation intelligence based on ILP for feature selection of a project. Proposed approach was refined on the basis of the following, i.e., modeling, exploration, and consolidation.

Ngo-The and Ruhe [49] proposed an evolutionary approach EVOLVE^{+}, which is an extension of EVOLVE*, by adding "*soft constraints and objectives into decision making process*". In the proposed method, ELECTRE IS was used to handle the uncertainty in preference relation. A feature or a software requirement can be considered as a part of release, if it is ready before the release date. Ngo-The and Ruhe [50] proposed a two phase optimization approach called "OPTIMIZE$_{RASORP}$" by combining the ILP and Genetic Programming. The proposed method was evaluated on 600 randomly generated problems with varying parameters.

Sagrado *et al.* [19] developed a tool called CARE, i.e., Case Aided Requirement, to guide the decision maker so that best set of requirements can be selected in the next release of the software. Simulated Annealing, Genetic Algorithm and ACO were integrated in a requirement tool for the selection of requirements.

Al-Emran *et al.* [51] presented an integrated method by combining "*Monte-Carlo simulation with process simulation*", i.e., "*Project Simulation for Operational Release Planning*" (ProSim/ORP). Four uncertainty factors were investigated, i.e., " (i) the number of new features arriving during release construction, (ii) the estimated effort needed to implement features, (iii) the availability of developers, and (iv) productivity of developers". As a result, authors found that "*the number of new features arriving during release construction*" is the important uncertainty factor. In another study, Al-Emran *et al.* [52] proposed a method called DECIDE$_{RELEASE}$ and it uses "*simulation based analysis*" (SBA) and "*multi-criteria decision analysis*" (MCDA). SBA and MCDA were applied on top of the existing strategic release planning. DECIDE$_{RELEASE}$ was proposed to explore the "*robustness of the operational plans of upcoming releases*".

Sagrado *et al.* [53] proposed a method by integrating the GRASP, GA, and Ant Colony System (ACS) for the selection of software requirements considering the interaction among requirements. They have tested each algorithm on two sets of data after performing 100 runs for each of the data sets. After analyzing the algorithm, they found that GRASP and ACS have the same solution but GRASP has better execution time than ACS. The solution found by the GA was not satisfactory. On the second dataset GRASP was better than ACS in terms of satisfaction; and GA again obtain the worst solution.

Kumari *et al.* [54] proposed a method for the selection of software requirements using "*Quantum Inspired Multi-objective Differential Evaluation Algorithm (QMDEA)".* In the same year, Kumari *et al.* [55] proposed another algorithm for SRS using "*Quantum-Inspired Elitist Multi-objective Differential Evolution Algorithm* (QEMEA)". In another study, Kumari and Srinivas [56] proposed a "*Multi-objective Quantum Inspired Hybrid Differential Evolution (MQHDE)"* for SRS. The proposed MQHDE was tested on banking project and the results were compared with the NSGA-II. In a similar study, Kumari *et al.* [57] compared the results of MQHDE with NSGA-II and MoCell on the basis of six benchmark problems; these problems were adopted from [3].

Cai and Wei [58] presented an integrated method of "*domination and decomposition based multi-objective evolutionary*" algorithm (MOEA/DD) for the selection of software requirements. Authors analyzed the "*decomposition based approach and compared it with domination based*" approach on SRS problem. At the time of constructing the subpopulation of the sub-problem, density based mechanism was used to switch between domination and decomposition. Finally, authors show that MOEA/DD outperforms other approaches. Cheng *et al.* [59] proposed a Memtic algorithm to address the NRP. In this method SA is integrated with MOEA/D as local search engine. The results of the proposed method were compared with multi-start SA, genetic algorithm, backbone based algorithms, NSGA-II, and MOEAD.

In 2016, Kumari and Srinivas [60] compared the following algorithms, i.e., *QEMEA*, *MQHDE*, and *QMDEA* for the solution of SRS problem with the NSGA-II. On the basis of the comparative study, it was found that MQHDE produces high quality solution and QMDEA produces well distributed solutions with extreme boundary solutions.

Fuchshuber and Barros [61] visualize the SRS problem using landscape visualization. They have applied visualization pattern along with the hill climbing algorithm to improve the heuristics for the selection of software requirements. After experimental work they show that proposed method generates better results than the original hill climbing.

Sadiq and Afrin [62] proposed a method for the SRS by integrating the fuzzy AHP and fuzzy "*Technique for Order Preference by Similarity to Ideal Solutions*" TOPSIS methods. Proposed method was applied for the selection of the requirements of "*Institute Examination System*".

Aydemir *et al.* [63] revisited the NRP using goal based approach and proposed a next release tool (NRT). This tool supports the goal based modeling and reasoning for the selection of goals and requirements. Cai *et al.* [64] proposed two multi-objectives Memtic algorithms (MOMA) using two different adaptive mechanisms for the solution of software release problem and travelling salesman problem. In the proposed method, SA is integrated as a local refinement process. The adaptive Memtic algorithm that uses utility is named as uMOMA-SA; and the algorithm which uses external archive as a local search is called aMOMA-SA.

Zhang et al. [65] proposed an integrated method using 2 phase external archive and MOEA/D, i.e., 2EAG-MOEA/D. In this method evolutionary process is divided into two phases, i.e., "*convergence and diversity phase*". The results of the proposed methods were compared with NSGA-II, MOEA/D, etc.

Pitangueira et al. [66] reformulated the MONRP in which different stakeholder's participated during the selection of software requirements; and proposed a risk –aware multi-stakeholder NRP using satisfiability modulo theory (SMT). Two important SMT solvers were used to solve the MONRP, i.e., Yices and Z_3.

Table 9: Integrated methods

Paper ID	Authors	Methods	Applications/ Experimentation/ Dataset	Evaluating Criteria/ Objectives
I-1	Ruhe and Ngo-The [47]	Evolutionary computing and MCDA are integrated with human intelligence	Illustrated by a simple case study	Time, benefit, and quality
I-2	Ruhe and Saliu [48]	ILP is integrated with heuristics to generate the set of good solutions	Tested on 15 features/ requirements	Effort, Cost, and Time
I-3	Ngo-The and Ruhe [49]	Evolutionary computing and ELECTRE IS	Bases on real world example	Cost, Benefit, and Quality
I-4	Ngo-The and Ruhe [50]	ILP is integrated with genetic algorithm	Evaluated on 600 randomly generated problems	
I-5	Sagrado *et al.* [19]	Simulated annealing, Genetic algorithms , and ACO	Only a tool is proposed and not validated on any kind of dataset	Cost and Customer satisfaction
I-6	Al-Emran *et al.* [51]	Monte-Carlo simulation with process simulation	Time and uncertainty	Case study at Chartwell

				Technology Inc.
I-7	Al-Emran *et al.* [52]	Simulation based analysis, and ELECTRE IS	Effort and productivity	Case study with 20 requirements and 6 developers
I-8	Sagrado *et al.* [53]	GRASP, GA, and ACS	Apply the method on two data sets	Effort and Time
I-9	Kumari *et al.* [54]	Quantum computing combined with differential evolutionary algorithm	Tested on six benchmark problems, adopted from Zhang *et al* (2007)	Cost and Degree of importance of customers
I-10	Kumari *et al.* [55]	Quantum computing combined with differential evolutionary algorithm	Tested on six benchmark problems, adopted from Zhang *et al* (2007)	Cost and Degree of importance of customers
I-11	Kumari and Srinivas [56]	Quantum computing combined with evolutionary algorithm, and genetic algorithm	Tested on banking project	Cost and Degree of importance of customers
I-12	Kumari *et al.* [57]	Quantum computing combined with evolutionary algorithm, and genetic algorithm	Tested on six benchmark problems, adopted from Zhang *et al* (2007)	Cost and Degree of importance of customers
I-13	Cai and Wei [58]	Decomposition and Domination based methods	Tested on 30 Customers and 303 requirements, 50 Customers and 200 requirements, 50 Customers and 500 requirements, 80 Customers and 800 requirements, 100 Customers and 1000 requirements, and 120 Customers and 1200 requirements	Cost and Customer satisfaction
1-14	Cheng *et al.* [59]	Simulated annealing and MOEA/D	Use open source software projects of Eclipse and Gnome.	Cost and Profit
I-15	Fuchshuber and Barros [61]	Landscape visualization with Hill Climbing	Tested on random data	Cost and Profit
I-16	Kumari and Srinivas [60]	Quantum computing was combined with evolutionary algorithm and GA	Tested on six benchmark problems, adopted from Zhang *et al* (2007)	Cost and Degree of importance of customers
I-17	Paixao and Souza [69]	Scenario based robust optimization framework, simulated annealing and GA	Tested on both artificial and real world instances extracted from Eclipse and Mozilla	Cost and importance of requirements
I-18	Paixao and Souza[70]	Scenario based robust optimization framework	Synthetic and real world instances	Cost and importance of

		and GA		requirements
1-19	Li *et al.* [72]	Monte-Carlo simulation and search based optimization	Motorola dataset having 35 requirements	Cost, revenue, & Uncertainty
I-20	Harman *et al.* [73]	Nemhauser-Ullmann's algorithm for sensitivity analysis and One-at-a time (OAT)	Motorola dataset having 35 requirements	Cost and Revenue
I-21	Paixao and Souza [78]	Robust optimization framework, simulated annealing, and GA	Synthetic and real world dataset	Cost and importance of requirements
I-22	Aydemir *et al.* [63]	Goal based Modeling and Reasoning with Satisfiability Modulo Theorem	Air Traffic Management	Cost and customer value contribution
I-23	Cai *et al.* [64]	Simulated Annealing is Integrated with Multi-objective Memetic Framework	Both synthetic and real world dataset	Cost and importance of requirements
I-24	Zhang *et al*. [65]	External archive is integrated with MOEA/D	Data extracted from external archive	Cost and importance of requirements
I-25	Pitangueira *et al.* [66]	Risk, Multiple-stakeholders, and Satisfiability modulo theory solver	Microsoft word with 50 requirements and 4 stakeholders; ReleasePlanner^{TM} with 25 requirements and 9 stakeholders.	Cost, average value, and risk
I-26	Sadiq and Afrin [62]	Fuzzy AHP and Fuzzy TOPSIS	Institute Examination System	Security, Reliability, Economy

5. Observations and Recommendations

SRS and software requirements prioritizations are two different issues in the area of software requirements engineering [1]. The objective of SRS is to decide whether to include the requirements in the next release of the software or exclude it. On the other side, in requirements prioritization the priorities of the requirements are computed [67, 68]. It is observed that there is a growth of studies related to SRS in the year of 2015 and 2014, as shown in Fig. 2. It is estimated that number of publications will keep increasing in the year because of the importance of the SRS in the area of software requirements engineering. It is also observed that both "*synthetic and real world instances*" have been used for experimental work. In the following sub-section, we discuss our research issues:

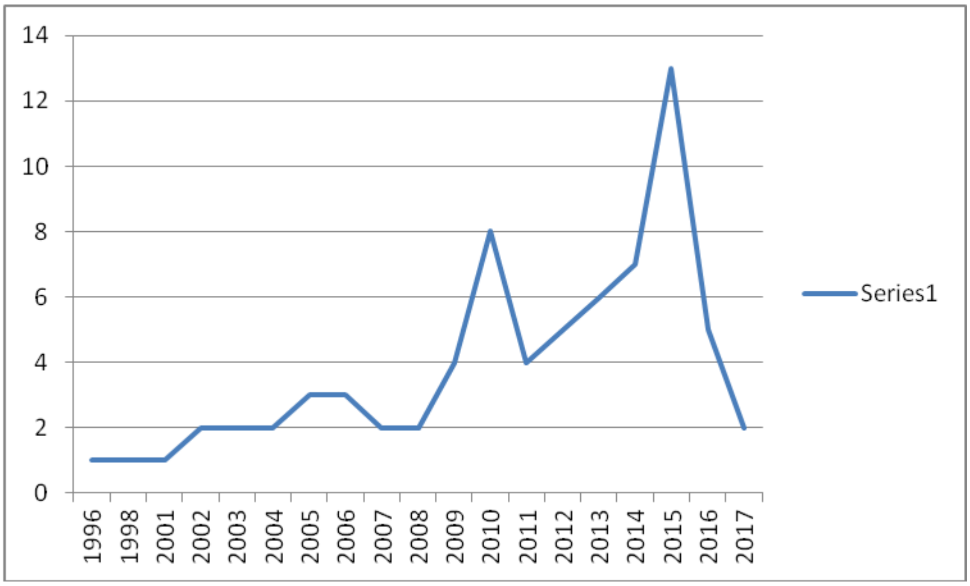

Figure. 2. Number of papers by year of publications

5.1 Existing methods for SRS (RQ-1)

The first objective of this paper is to find out the existing methods adopted in the selection of software requirements. We have identified following methods which is used for the SRS: ILP, ACO, GA, Artificial Bee Colony (ABC), "*Teaching Learning Based Optimization*" (TLBO) Algorithm, "*Differential Evolution*" (DE) Algorithm, "*Backbone-based Multilevel*" Algorithm (BMA), AHP, Greedy Algorithm, Hill Climbing Algorithm, SA, NSGA-II, Pareto GA, MoCell, PAES, SPEA-II, IWO/MO, "*Decomposition and Domination based Evolutionary*" Algorithm, Memetic Algorithm, AGORA, and Quality Function Deployment. There are some other methods which have been used in integrated methods like Monte Carlo Simulation (MCS), Simulation based analysis, MCDA algorithm (ELECTRE IS), Landscape Visualization, Nemhauser-Ullmann's exact optimization algorithm, Goal Based Approaches, and Satisfiability Modulo Theory. In our study, we find out that Individual methods and integrated methods are equally important than the group of methods. On the basis of our analysis, we identify that the most popular individual method for the solution of SRS problem are ILP followed by ACO, GA, BMA, AHP, etc. ILP has attracted more attention in the area of SRS because it provides the exact solution for the small set of data. Bagnall *et al.* [2] consider ILP as one of the method for the solution of NRP with the other methods, i.e., Greedy algorithm, Hill climbing, and SA. After ILP, ACO has also received much attention to select the requirements because it is an efficient method to deal with the dynamic applications. As shown in Table 8 and Table 9, group of methods and integrated methods have also been used in the area of SRS. It was noticed that NSGA-II and SA are the most popular methods that have been used for the experimental work during the selection of software requirements.

5.2 Most popular criteria (RQ-2)

The second objective of this paper is to identify the most popular criteria used for the selection of software requirements. On the basis of our study, we have identified following criteria that have been used for the SRS, i.e., importance of requirements, cost, estimated revenue, development time, on time delivery, profit, effort, risk, customer satisfaction, benefits, risk factors, goal and vision fit, organizational politics, licensing arrangements, financial conditions, implementation and service ability, consulting service, R& D technology, vender reputation, training and support, market trends, replacibility, instantiability, conformance, adaptability, testability, stability, changeability, analyzability, time behavior, resource behavior, understandability, operability, learnability, recoverability, maturity, fault tolerance, suitability, security, interoperability, compliance, and accuracy. Among the listed criterions, the most popular criterion is the cost followed by the importance of requirements, time, revenue, and risk.

5.3 Fuzzy based methods (RQ-3)

The third objective of this paper is to identify the number of studies in which fuzzy based methods have been used. Based on our analysis, we identify that fuzzy based methods have received less attention for the selection of software requirements. We have identified two studies which support fuzzy based methods. For example, Sadiq and Jain [36] applied fuzzy preference relation for the selection of goals/ requirements in goal oriented method. In another study, Sen and Baracli [5] used fuzzy based quality function deployment methodology for the selection of software requirements.

5.4 Uncertainty in SRS problem (RQ-4)

Uncertainty is an inherent characteristic in the development of software project. We have identified seven studies (10.29 %) which address the uncertainty issues during the selection of software requirement.

Paixao and Souza [69] formulated the NRP by using scenarios; and proposed a robust optimization framework for the production of robust solutions. To measure the price of robustness, three experiments were designed by using the simulated annealing and genetic algorithm. Proposed robust formulation considers the uncertainties presented in the input variables. In another study, Paixao and Souza [70] proposed a "*recoverable approach for the NRP*". Paixao and Souza [71] reformulate the NRP using robust optimization framework in the presence of uncertainties.

Li *et al.* [72] adopted the "*search –based optimization technique with Monte-Carlo simulation (MCS) to address the uncertainty and risk*" during SRS. They have proposed two notions of the uncertainty measurements for NRP, i.e., MCS for NRP- uncertainty size (MCNRP-US) and MCS for NRP-risk (MCNRP-R).

Herman *et al.* [73] coined the requirements sensitivity analysis (RSA) problem for ordering the requirements attributes according to their impact on the choice of their requirements for the next release. To propose an exact scalable solution, one-at-a-time (OAT) solution using the "*Nemhauser-Ullmann's exact optimization*" algorithm was used. Authors developed a tool, i.e., OATSAC, for cost-benefit sensitivity analysis so that contribution to the overall uncertainty of a solution can be assessed.

Li [74] proposed a "*decision support framework for analyzing the uncertainty in requirements selection and optimization*". This work was extended by Li et al. [75] in which a NRP decision analysis framework, called METRO, was developed by including the Monte-Carlo Simulation to capture the requirements uncertainty.

5. 5 Limitations (RQ-5)

The last objective of this paper is to identify the limitations of the existing approaches. In this section we mainly focus on ILP and ACO because these are the popular methods used in the selection of software requirements. ILP has been applied to NRP but the scalability issue has not been explored except Harman *et al.* [73] and Xuan *et al.* [34]. ILP method is suitable to develop the static models for the SRS but in practical life, requirements evolve according to the need of stakeholders, market requirements, etc. In such cases, ILP does not support the dynamic nature of software requirements. ILP method is limited to only two variables but in practical life more than three objectives could be there in SRS problem. ACO is an effective method to solve NP hard problems like, NRP. To improve the performance of ACO algorithm, different variations of hill climbing are used. For example, first found hill climbing operator was used into AC to improve the quality of the solutions. Using ACO, it is uncertain to converge the solution and difficult to estimate the theoretical speed of the convergence. Existing ant colony based methods for the selection of software requirements do not support the following variations of ACO, i.e., (i) elitist ant system, (ii) max-min ant system, (iii) rank based ant system, (iv) continuous orthogonal ant system, and (v) recursive ant colony optimization. Most of the SRS methods do not support stakeholder identification and their communication. In real life applications, several stakeholders participate at the time of SRS process. Therefore, the lack of the stakeholder participation can lead to the generation of the vague results. Only few studies have considered stakeholder identification and communication [36, 37, 38] during SRS process. In real life applications, stakeholders may use linguistic values like good, very good, low, etc., instead of 1, 2, or 3, in order to specify his/her preferences during SRS process. Fuzzy based approaches have also received less attention in the area of SRS. Most of the methods of SRS are limited to only one or two objectives. In practical situations, more than two objectives may be present in SRS problem. Therefore, there is a need to use other objectives like performance, reliability, usability, security, and maintainability at the time of SRS process.

6. Conclusions and Future Work

This paper presents a literature review of SRS methods from January 1996 - December 2017. In this paper, firstly, it was found that various individual and integrated methods were proposed to solve the SRS problem. Several groups of methods have also been used for the selection of software requirements. The most popular methods are ILP, ACO, Simulated Annealing, and NSGA-II. Secondly, the most popular criteria for the selection of the software requirements are cost followed by efforts and time. Third, we identify that in SRS methods less attention is given to fuzzy based approaches. In our work, we also observe that requirements uncertainty is considered as an important issue during SRS. Finally, we discuss the limitations in the existing methods and identify some research issues in the area of SRS for future work; and it is given below:

1. Few SRS methods have proposed integrated methods by using quantum computing in evolutionary algorithm. There is lack of quantum–inspired genetic (QIG) algorithm and quantum–inspired immune clonal (QIIMC) algorithms for the selection of software requirements. In future, QIG and QIIMC methods can be used to solve the SRS problem and results can be compared with the quantum inspired evolutionary algorithms, NSGA-II, and simulated annealing.
2. Different Metaheuristics methods have been used for the selection of software requirements, for example, PSO, ACO, GA, Memetic algorithms, simulated annealing, GRASP, DE, TLBO algorithm, ABC algorithm. In future, other metaheuristic algorithm may be used for SRS, i.e., cellular automata, immune systems, harmony search, Cuckoo search, bacterial foraging optimization algorithms, and biography based optimization algorithms.
3. Fuzzy based approaches may be integrated with ILP and Metaheuristics methods for the selection of software requirements. Multicriteria decision making (MCDM) method like, AHP and "*technique for order of preference by similarity to ideal solutions"* (TOPSIS) may be integrated with ILP and metaheuristic algorithm for the selection of software requirements.
4. Most of the focus in the area of SRS is given to the SBSE. Therefore, there is a need to develop the algorithms for SRS by using the MCDM methods like, AHP, TOPSIS, ELECTRE, PROMETHEE, etc.

REFERENCES

[1]. Petersen. K., and Ali. N. B, "Operationalizing the Requirements Selection Process with Study Selection Procedures from Systematic Literatures Surveys," in Proceedings of the 6th International Workshop on Requirements Prioritization and Communication, Essen, Germany, 2015, pp. 102-113.

[2]. Bagnall. A. J, Rayward –Smith. V. J and Whittley. I. M, “The Next Release Problem”, Information and Software Technology, Vol. 43, No. 14, pp. 883-890, 2001.
[3]. Zhang. Y, Harman. M and Mansouri. S.A, “The Multi-Objective Next Release Problem,” in Proceedings of the ACM 9th Annual Conference on Genetic and Evolutionary Computation, 2007, pp. 1129-1137.
[4]. Karlsson. J and Ryan K, “Supporting the Selection of Software Requirements,” in Proceedings of the IEEE 8th International Workshop on Software Specification and Design, 1996, pp. 146-149.
[5]. Sen. S.G and Baracli. H, “Fuzzy Quality Function Deployment Based Methodology for Acquiring Enterprise Software Requirements Selection,” Expert Systems with Applications, Vol. 37, pp. 3415-3426, 2010.
[6]. Moon. S.Y, Jeon. B and Kim. Y.C, “Validation of the Cost Estimation with Object Oriented Function Point (FP) through Software Visualization”, Journal of Engineering Technology, Vol.7, 246-261, Oct 2018.
[7]. Sadiq. M, Ahmad. J and Mohammad C. W, “ Elicitation of Security Requirements from the Selected Set of Functional Requirements using Fuzzy Based Approach”, Journal of Engineering Technology, 2019 (in Press).
[8]. Pitangueira. A. M, Maciel. R. S. P and Barros. M, “Software Requirements Selection and Prioritization using SBSE Approaches: A Systematic Review and Mapping of the Literature,” Journal of Systems and Software, Vol. 103, pp. 267-280, 2015.
[9]. Pitangueira. A. M, Maciel. R. S. P, Barros. M. O and Andrade. A. S, “A Systematic Review of Software Requirements Selection and prioritization using SBSE Approaches,” in Proceedings of Search Based Software Engineering, Volume 8084 of the Series Lecture Notes in Computer Science, Springer, 2013,188-208.
[10]. Jung. H. W, “Optimizing value and cost in requirements analysis,” IEEE Software, Vol. 15, No. 4, (1998), 74-78.
[11]. Van Den Akker. J. M, Brinkkemper. S, Diepen. G and Versendaal. J, “Determination of the Next Release of Software Product: An Approach using Integer Linear Programming,” in Proceedings of 11th International Workshop on Requirements Engineering Foundation for Software Quality, (2005), 119-124.
[12]. Van den Akker. J. M, Brinkkemper. S and Diepen. G, “Flexible Release-Planning using Integer Linear Programming,” in Proceedings of 11th International Workshop on Requirements Engineering for Software Quality, (2005), 247-262.
[13]. Van den Akker. J.M, Brinkkemper. S, Diepen. G and Versendaal. J, “Software Product release Planning through Optimization and What-if Analysis,” Information and Software Technology, Vol. 50, No. 1-2, (2008), 101-111.
[14]. Li. C, Van Den Akker. M, Brinkkemper. S and Diepen. G, “An Integrated Approach for Requirements Selection and Scheduling in Software Release Planning,” Requirements Engineering, Springer, Vol. 15, (2010), 375-396.
[15]. Sureka. A, “Requirements Prioritization and next release Problem under Non-additive Value Candidates,” in Proceedings of the 23rd IEEE Australian Software Engineering Conference, (2014), 120-123.
[16]. Veerapen. N, Ochoa. G, Harman. M and Burke. E. K, “An Integer Linear Programming Approach to the Single and Bi-Objective Next Release Problem,” Information and Software Technology, Vol. 65, No. C, (2015), 1-13.
[17]. Mougouei. D, “Factoring Requirements Dependencies in Software Requirements using Graph and Integer Programming,” in Proceedings of International Conference on Automated Software Engineering, (2016), 884-887.
[18]. Sagrado. J., Aguila. I. M, “Ant Colony Optimization for Requirements Selection in Incremental software Development,” in Proceedings of the 1st IEEE International Symposium on Search Based Software Engineering. Cumberland Lodge, Windsor, UK, 2009.
[19]. Sagrado. J, Aguila I. M and Orellana F. J, “Ant Colony Optimization for the Next Release Problem: A Comparative Study,” in Proceedings of the 2nd IEEE International Symposium on Search Based Software Engineering, (2010), 67-76.
[20]. Jiang. H, Zhang. J, Xuan. J, Ren. Z and Hu. Y, “A Hybrid ACO Algorithm for the Next Release Problem,” in Proceedings of the 2nd International Conference on Software Engineering and Data Mining, (2010), 166-171.

[21]. Souza. J. T, Maia. C. L. B, Ferreira. T. N, Carmo. R. A. F and Brasil M. M. A, "An Ant Colony Optimization Approach to the Software Release Planning with Dependent Requirements," in Proceedings of Search Based Software Engineering, LNCS, Springer, Vol. 6959, (2011), 142-157.
[22]. Chaves-Gonzalez. J. M, Perez-Toledano M. A and Navasa A, "Software Requirements Optimization using Multi-Objective Swarm Intelligence Evolutionary Algorithm," Knowledge Based System, Elsevier, Vol. 83, No. C, (2015), 105-115.
[23]. Chaves-Gonzalez. J. M, Perez–Toledano. M. A and Navasa. A, "Teaching Learning Based Optimization with Pareto Tournament for the Multi-Objective Software Requirements Selection" Engineering Applications of Artificial Intelligence, Elsevier, Vol. 43, (2015), 89-101.
[24]. Ranjith. N and Marimuthu. A, "A Multi-Objective Teacher-Learning-Artificial –Bee Colony Optimization for Software Requirements Selection," Indian Journal of Science and Technology, Vol. 9, No. 4, (2016), 1-9.
[25]. Sagrado. J, Del Aguila. I. M, Orellana. F. J and Tunez. S, "Requirements Selection: Knowledge Based Optimization Techniques for Solving the Next Release Problem," in Proceedings of 6th Workshop on Knowledge Engineering and Software Engineering at 33rd German Conference on Artificial Intelligence, (2010), 40-51.
[26]. Ferreira T. N, Araujo A. A, Neto. A. D. B and Souza. J. T, "Incorporating User Preferences in Ant Colony Optimization for the Next Release Problem," Applied Soft Computing, Elsevier, Vol. 49, (2016), 1283-1296.
[27]. Ruhe. G and Greer. D, Quantitative Studies in Software Release Planning Under Risk and Resources Constraints, in Proceedings of IEEE International Symposium on Empirical Software Engineering, (2003), 262-270.
[28]. Greer. D and Ruhe. G, "Software Release Planning: An Evolutionary and Iterative Approach," Information and Software Technology, Vol. 46, No. 4, (2004), 243-253.
[29]. Araujo. A. A and Paixao. M.H.E, "Machine Learning for User Modeling in An Interactive Genetic Algorithm for Next Release Problem," in Proceedings of the 6th International symposium on Search Based Software Engineering, LNCS, Springer, Vol. 8636, (2014), 228-233.
[30]. Silva. T. G. N, Rocha. L. S and Maia. J.E.B, "An Effective Method for MOGAs Initialization to Solve the Multi-Objective Next Release Problem," in Proceedings of the 13th Mexican International Conference on Artificial Intelligence, LNCS, Springer, (2014), 25-37.
[31]. Chaves-Gonzalez. J. M and Perez- Toledano. M. A, "Differential Evolution with Pareto Tournament for the Multi-Objective Next Release Problem," Applied Mathematics and Computation, Elsevier, Vol. 252, (2015), 1-13.
[32]. Jiang. H, Xuan. J and Ren. Z, "Approximate Backbone Based Multilevel Algorithm for the Next Release Problem," in Proceedings of the 12th Annual Conference on Genetic Evolutionary Computation, (2010), 1333-1340.
[33]. Lundy. M and Mees. A, "Convergence of An Annealing Algorithm," Mathematical Programming, Elsevier, Vol. 34, No. 1, (1986), 111-124.
[34]. Xuan. J, Jiang. H, Ren. Z and Luo. Z, "Solving the Large Scale Next Release Problem with a Backbone Based Multilevel Algorithm," IEEE Transactions on Software Engineering, Vol. 38, No. 5, (2012), 1195-1212.
[35]. Ruhe.G., Eberlein. A., and Pfahl. D., "Trade-off Analysis for Requirements Selection," International Journal of Software Engineering and Knowledge Engineering, Vol. 13, No. 4, pp. 345-366, 2003.
[36]. Sadiq. M and Jain. S. K, "A Fuzzy Based Approach for the Selection of Goals in Goal Oriented Requirements Elicitation Process," International Journal of Systems Assurance Engineering and Maintenance, Springer, Vol. 6, Issue 2, (2015), 157-164.
[37]. Sadiq. M, "A Fuzzy-Set Based Approach for the Prioritization of Stakeholders on the Basis of the Importance of Software Requirements", IETE Journal of Research, Vol. 63, Issue 5, (2017), 616-629, 2017.
[38]. Pitangueira. A.M, "Incorporating Preferences from Multiple Stakeholders in Software Requirements Selection: An Interactive Search Based Approach," in Proceedings of the 23rd IEEE International Requirements Engineering Conference, (2015), 382-287.
[39]. Zhang. Y, Harman. M, Finkelstein. A and Mansouri S.A, "Comparing the Performance of Metaheuristic for the Analysis of Multi-Stakeholder Trade-Offs in Requirements Optimization," Information and Software Technology, Vol. 53, No. 7, (2011), 761-773.

[40]. Kaiya. H, Horai. H and Saeki. M, "AGORA: Attributed Goal-Oriented Requirements Analysis Method," in Proceedings of the IEEE International Conference on Requirements Engineering, (2002), 13-22.
[41]. Baker. P, Harman. M, Steinhofel. K and Skaliotis. A, "Search Based Approaches to Component Selection and Prioritization for the Next Release Problem," in Proceedings of the 2nd IEEE International Conference on Software Maintenance, (2006),176-185.
[42]. Durillo J. J., Zhang Y., Alba E., Nebro A. J., "A Study of Multi-Objective Next Release Problem," in Proceedings of 1st International Symposium on Search Based Software Engineering, 2009, pp. 49-58.
[43]. Durillo J. J., Zhang Y., Alba E., Harman M., and Nebro A. J., "A Study of the Bi-Objective Next Release Problem," Empirical Software Engineering, Vol. 16, No. 1, pp. 29-60, 2011.
[44]. Cai. X, Wei. O and Huang. Z, "Evolutionary Approaches for Multi-Objective Next Release Problem" Computing and Informatics, Vol. 31, (2012), 847-875.
[45]. Mausa. G, Grabac. T. G, Basic. B. D and Pavcevic. M. O. "Hill Climbing and Simulated Annealing in Large Scale Next Release Problem," in Proceedings of IEEE EUROCON, (2013), 52-459.
[46]. Botelho. G, Rocha. A, Britto. A and Silva. L, "Investigating Bio-Inspired Strategies to Solve Large scale Next Release Problem" in Proceedings of CIBSE Conference and Exhibition, (2015), 1-14.
[47]. Ruhe. G and Ngo-The. A, "Hybrid Intelligence in Software Release Planning," International Journal of Hybrid Intelligence Systems, Vol. 1, No. 1-2, (2004), 99-110.
[48]. Ruhe. G and Saliu. M. O, "The Art and Science of Software Release Planning," IEEE Software, (2005), 47-53.
[49]. Ngo-The. A. and Ruhe. G., "A Systematic Approach for Solving the Wicked Problem of Software release Planning", Soft Computing, Springer, Vol. 12, (2008),95-108.
[50]. Ngo-The A. and Ruhe G., "Optimized Resource Allocation for Software Release Planning," IEEE Transactions on Software Engineering 2009; 35(1): 109-123.
[51]. Al Emran. A, Kapur. P, Pfahl. D and Ruhe. G, "Studying the Impact of Uncertainty in Operational Release Planning-An Integrated Method and its Initial Evolution," Information and Software Technology, Vol. 52, No. 4, (2010), 446-461.
[52]. Al Emran. A, Kapur. P, Pfahl. D and Ruhe. G, "Decision Support for Product Release Planning Based on Robust Analysis," in Proceedings of the 18th International Requirements Engineering Conference, (2010), 157-166.
[53]. Sagrado J.D, Aguila. I. M and Orellana. F.J, "Requirements Interaction in the Next Release Problem," in Proceedings of GEECO, (2011), 241-242.
[54]. Kumari. A. C, Srinivas. K and Gupta. M. P, "Software Requirements Selection Using Quantum-Inspired Multi-Objective Differential Evolution Algorithm," in Proceedings of the CSI 6th International Conference on Software Engineering, (2012), 1-8.
[55]. Kumari. A. C, Srinivas. K and Gupta. M. P, "Software Requirements Selection Using Quantum-Inspired Elitist Multi-Objective Differential Evolution Algorithm," in Proceedings of the IEEE International Conference on Advances in Engineering, Science, and Management, (2012), 782-787.
[56]. Kumari. A. C, Srinivas. K and Gupta. M. P, "Software Requirements Optimization Using Multi-Objective Quantum-Inspired Hybrid Differential Evolution" in Proceedings of EVOLVE-A Bridge between Probabilities Set Oriented Numerics and Evolutionary Computation II, 175, AICS, Springer, (2013), 107-120.
[57]. Kumari. A. C and Srinivas. K, "Search Based Software Requirements Selection: A Case Study," International Journal of Computer Applications, Vol. 64, No. 21, (2013), 28-34.
[58]. Cai. X and Wei. O, "A Hybrid of Decomposition and Domination Based Evolutionary Algorithm for Multi-Objective Software Next Release Problem," in Proceedings of the 10th IEEE International Conference on Control and Automation (ICCA), (2013), 412-417.
[59]. Cheng. X, Huang. Y, Cai. X and Wei. O, "An Adaptive Memetic Algorithm Based on Multi-Objective Optimization for Software Next Release problem," in Proceedings of the ACM Genetic and Evolutionary Computing Conference, (2014), 185-186.
[60]. Kumari. A. C and Srinivas. K, "Comparing the Performance of Quantum –Inspired Evolutionary Algorithm for the Solution of Software Requirements Selection Problem," Information and Software Technology, Vol. 76, (2016), 31-64.
[61]. Fuchshuber. R and Barros. M. O, "Improving Heuristic for the Next Release Problem Through Landscape Visualization," in Proceedings of the 6th International Conference on Search Based Software Engineering, LNCS, Springer, Vol. 8636, (2014), 222-227.

[62]. Sadiq M. and Afrin A., "An Integrated Approach for the Selection of Software Requirements Using Fuzzy AHP and fuzzy TOPSIS Methods," IEEE International Conference on Intelligent Computing, Instrumentation, and Control Technologies, (2017), 1094-1100.
[63]. Aydemir. F. B, Mekuria. D. N, Giorgini. P and Mylopoulos. J, "Next Release Tool," in Proceedings of the 34th International Conference on Conceptual Modeling, (2015), 1-4.
[64]. Cai. X, Cheng. X, Fan. Z, Goodman. E and Wang L, "An Adaptive Memetic Framework for Multi-Objective Combinatorial Optimization Problems: Studies on Software Next Release and Travelling Salesman Problems," Soft Computing, Springer, Vol. 21, Issue 9, (2017), 2215-2236.
[65]. Zhang. N, Huang. Y and Cai. X, "A Two-phase External Archive Guided Multi-Objective Evolutionary Algorithm for the Software Next Release Problem," in Proceedings of Bio-Inspired Computing-Theories and Applications, Volume 562 of the Series, Communications in Computer and Information Science, Springer, (2015), 664-675.
[66]. Pitangueira. A. M, Tonella. P, Susi. A, Maciel. R. S and Barros. M, "Risk Aware Multi-Stakeholder Next Release Planning Using Multi-objective Optimization," in Proceedings of Requirements Engineering: Foundations for Software Quality, Volume 9619 of the Series Lecture Notes in Computer Science, Springer, (2016), 3-18.
[67]. Achimugu. P, Selamat. A., Ibrahim. R and Mahrin. M. N, "A Systematic Literature Review of Software Requirements Prioritization Research," Information and Software Technology, Vol. 56, (2014), 568-585.
[68]. Behzadian. M, Otaghsare. S. K, Yazdani. M and Ignatius. J, "A State-of-the-art Survey of TOPSIS applications," Expert Systems with Applications, Vol. 39, (2012) 13051-13069.
[69]. Paixao. M. H. E and De Souza. J. T, "A Scenario Based Robust Model for the Next Release Problem," in Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, (2013), 1469-1476.
[70]. Paixao. M. H. E and De Souza. J.T, "A Recoverable Robust Approach for the Next Release Problem," in Proceedings of Search Based Software Engineering, Volume 8084 of the Series, Lecture Notes in Computer Science, Springer, (2013), 172-187.
[71]. Paixao. M and Souza. J, "A Robust Optimization Approach to the Next Release Problem in the Presence of Uncertainties," Journal of Systems and Software, Elsevier, Vol. 103, (2015) 281-295.
[72]. Li. L, Harman. M, Letier. E and Zhang. Y, "Robust Next Release Problem: Handling Uncertainty During Optimization," in Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, ACM, (2014), 1247-1254.
[73]. Harman. M, Krinke. J, Medina-Bulo. I, Palomo Lozano. F, Ren J and Yoo. S, "Exact Scalable Sensitivity Analysis for the Next Release Problem," ACM Transactions on Software Engineering and Methodology, Vol. 23, No. 2, (2014) 1-31.
[74]. Li. L, "Exact Analysis for Next Release Problem," in Proceedings of the IEEE International Conference on Software Engineering, (2016), 438-443.
[75]. Li. L, Harman. M, Wu. F and Zhang. Y, "The Value of the Exact Analysis in Requirements Selection", IEEE Transactions on Software Engineering, Vol. 43, Issue. 6, (2017), 580-596.
[76]. Li. C, Van Den Akker J. M, Brinkkemper. S and Diepen. G, "Integrated Requirements Selection and Scheduling for the Release Planning of a Software Product," in Proceedings of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality, 4542, LNCS, Springer, (2007), 93-108.
[77]. Sagrado. J, Aguila I. M and Orellana. F. J, "Multi-Objective Ant Colony Optimization for Requirements Selection," Empirical Software Engineering, Springer, Vol. 20, No. 3, (2015), 577-610.