Received January 7, 2018, accepted February 7, 2018, date of publication February 27, 2018, date of current version March 19, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2806944*

# Multi-Objective Optimum Solutions for IoT-Based Feature Models of Software Product Line

**ASAD ABBAS[1], (Student Member, IEEE), ISMA FARAH SIDDIQUI[2], SCOTT UK–JIN LEE[1], ALI KASHIF BASHIR[3], (Senior Member, IEEE), WALEED EJAZ[4], (Senior Member, IEEE), AND NAWAB MUHAMMAD FASEEH QURESHI[5]**

[1]Department of Computer Science and Engineering, Hanyang University ERICA Campus, Ansan 15588, South Korea
[2]Department of Software Engineering, Mehran University of Engineering and Technology, Jamshoro 76062, Pakistan
[3]Department of Science and Technology, University of the Faroe Islands, Tórshavn 100, Faroe Islands
[4]Department of Electrical and Computer Engineering, Sultan Qaboos University, Muscat 123, Oman
[5]Department of Computer Education, Sungkyunkwan University, Seoul 110-745, South Korea

Corresponding author: Scott Uk–Jin Lee (scottlee@hanyang.ac.kr)

This work was supported by the National Research Foundation of Korea through the Korean government (MSIP) under Grant NRF-2016R1C1B2008624.

**ABSTRACT** A software product line is used for the development of a family of products utilizing the reusability of existing resources with low costs and time to market. Feature Model (FM) is used extensively to manage the common and variable features of a family of products, such as Internet of Things (IoT) applications. In the literature, the binary pattern for nested cardinality constraints (BPNCC) approach has been proposed to compute all possible combinations of development features for IoT applications without violating any relationship constraints. Relationship constraints are a predefined set of rules for the selection of features from an FM. Due to high probability of relationship constraints violations, obtaining optimum features combinations from large IoT-based FMs are a challenging task. Therefore, in order to obtain optimum solutions, in this paper, we have proposed multi-objective optimum-BPNCC that consists of three independent paths (first, second, and third). Furthermore, we applied heuristics on these paths and found that the first path is infeasible due to space and execution time complexity. The second path reduces the space complexity; however, time complexity increases due to the increasing group of features. Among these paths, the performance of the third path is best as it removes optional features that are not required for optimization. In experiments, we calculated the outcomes of all three paths that show the significant improvement of optimum solution without constraint violation occurrence. We theoretically prove that this paper is better than previously proposed optimization algorithms, such as a non-dominated sorting genetic algorithm and an indicator-based evolutionary algorithm.

**INDEX TERMS** Software product line (SPL), feature modeling, Internet of Things (IoT), multi-objective optimization.

## I. INTRODUCTION

Software Product Line (SPL) is used intensively in software industry for development of families of software that share core common and variable functionalities. Each product of SPL differs from the others with variable features that provide functionalities according to end user requirements. Industry uses SPL to increase the reusability of features that reduce the development cost and time to market, which results in better product development [1], [2]. Development of SPL is based on two distinct processes: core development and application development. The first one is the process of developing common and variable features under the domain of SPL. The second one is the process of developing the product by using existing common and variable features in accordance to the stakeholder requirements [3]–[5]. Development of existing common and variable features consume cost and time in advance without any product derivation that can be remunerated by reusability in multiple products development [6], [7].

Feature Model (FM) is a tree structure which is used to manage the common and variable features of SPL. FM is a compact picture of all products under the domain of SPL where alternative, optional and OR group predefine constraints and relationships between features [8]. Development of the product is based on desired features selected from the FM, that fulfill the functional requirements and quality attributes of stakeholder [9]. Selection of features according to requirements of the stakeholder is a difficult, time consuming task in large FM configuration space, due to the complexity of relationships and constraints. During product configuration, the requirements from stakeholder compromises, do not satisfy on a single point such as lower memory consumption, lower cost and high performance. Therefore, SPL developers need to consider trade-off among inter-conflicting objectives [10], [11].

The Internet of Things (IoT) is used for technology advancement and is economically attractive in all sectors as a revolution of communication advancement [12]–[14] (e.g. transportation and health-care). IoT devices and applications enable the connectivity of different environments with respect to their context [15], [16] such as indoor and outdoor heat sensors. IoT is a paradigm that connects multiple internet things across different environments in the context of functional and non-functional requirements [17], [18]. Due to the importance of IoT in future applications, in this article, IoT application environments are being used to draw FMs. In the literature section, the contextual variability management of IoT applications by using feature modeling has already been discussed [19]. SPL is used to manage the contextual variability and to increase the reusability of IoT application features. However, selection of the best IoT application development features according to the end user objectives is a challenging task due to the existence of significant contextual variability in IoT environments. To satisfy the end user objectives, optimization is the best approach to achieve optimum features selection.

Optimization is a technique extensively used to find the optimum solutions for various problems in different engineering disciplines such as design engineering, and system engineering [20]–[22]. In literature, different multi-objective evolutionary algorithms have been used to find the optimum configuration from FM of SPL such as Indicator Based Evolutionary Algorithm (IBEA) and Non-dominated Sorting Genetic Algorithm (NSGA-II) [11], [23]. However, optimum solutions from these algorithms are not fully correct in the context of constraints violations occurrence. Moreover, none of these algorithms are feasible to acquire 100% correct optimum solutions of FM. To obtain fully correct optimum solutions from large and small FMs without constraint violation we adopted our previous proposed approach, the BPNCC algorithm, [24] that computes all possible combinations of SPL products without any constraint violations. In the BPNCC algorithm, all unique combinations were in binary form; selected features indicated by 1 and non-selected features indicated by 0.

In this paper, we have proposed Multi-Objective Optimum (MOO)-BPNCC approach to get the optimum solution for IoT applications without any relationship constraint violation. MOO-BPNCC is an extension of BPNCC and consists of three independent paths to acquire optimum solutions: 1) path A applies objective functions on all configurations for optimum solutions; however, this path increases space and time complexity on large FMs where millions of product configurations exist, 2) path B applies the objective functions on groups one by one and finds the optimum combinations from each group and then combines all groups optimum solutions, 3) path C reduces the complexity of FM by removing optional features that have constant values; 0 for minimization and 1 for maximization of objective functions. By using path C, time and space complexity can be reduced to achieve optimum solutions of FM. In BPNCC [24], we have already computed all possible solutions without any constraint violations; therefore, there is no possibility to miss any valuable solution for optimum combinations. In this study, we have found the minimized optimum solutions based on four minimized objective functions. We evaluated the outcomes of path A, B and C, found path C is giving the best performance. Furthermore, we have performed theoretical comparison of MOO-BPNCC with two well-known optimization algorithms: NSGA-II and IBEA from literature and concluded that path C of MOO-BPNCC performs better for optimum solutions.

Further, the paper is organized as follows: Section II is the Background, section III is the Related Work, section IV is the FM Multi-Objective Optimization, section V is the MOO-BPNCC of FM, section VI is the Experiment and Performance Evaluation, section VII is the Discussion and Limitations of MOO-BPNCC and section VIII is the Conclusion.

## II. BACKGROUND

FM is a user visible structure which represents complete information of all SPL products in terms of relationships and constraints among features. The hierarchy of all features in FM is composed by [25]:

- Relationships between features can be mandatory, alternative, optional and OR group
- Relationships of parent feature with child feature
- Constraints of features such as if feature A is selected, then feature B and C also must be selected or not selected

In attribute feature model, every feature contains functional and non-functional quality attributes. Based on quality attributes, features are selected for product derivation according to the user requirements [26]. Figure 1 shows the attribute feature model with four quality attributes: cost, performance, CPU and memory.

For product derivation, only terminal features are required, whereas, non-terminal features indicate the relationships between terminal features. In Fig. 1. the relationship between *inDays* and *Unlimited* is alternative, so end users must select one only. The end user must specify the functional and non-functional requirements and the objective functions maximize or minimize the variant product quality attribute.

## III. RELATED WORK

Loesch and Ploedereder [27] proposed variability optimization of SPL with the high complexity of feature relationships. Thousands of features make it difficult to manage the variability for product derivations according to the end

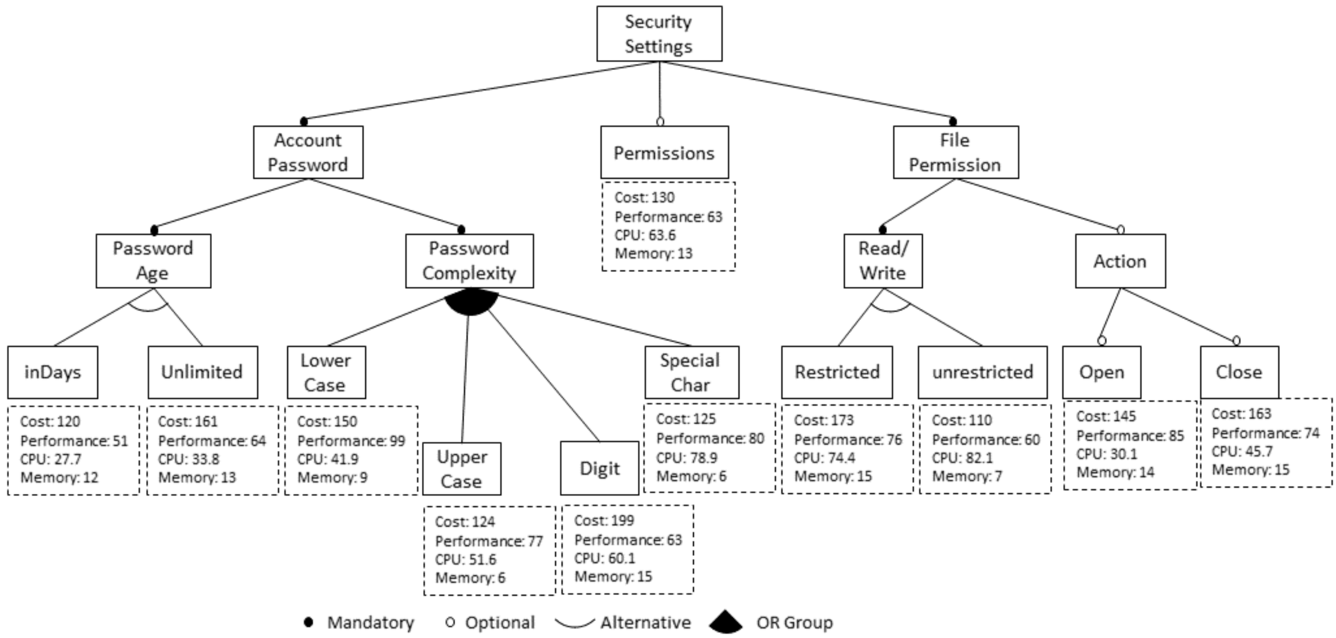


**FIGURE 1.** **IoT-based application attribute feature model.**

user perspective. The selection of variable features according to end user requirements such as cost and performance are difficult for a large number of features and relationship constraints. Authors have proposed the optimization method of feature selection by constructing a variable product-feature matrix that is used for final product configurations. Formal concept analysis has been used to extract the variable feature matrix, where common and variable features can be differentiated by finding the common features that are always used in every product. Optimization has been applied on obtained variable feature matrix with the attribute values of every feature in FM.

Guo et al. [28] presented Genetic Algorithm (GA) for optimized features selection of SPL FM. Minimization or maximization of objective functions is difficult to evaluate in FM when a large number of constraints and relationships exist. GA performs mutations and crossover on initial population, such as features combinations, and evaluates the objective function on each configuration to minimize or maximize the functions. The proposed approach is named as the Genetic Algorithm for optimized Feature Selection (GAFES) for SPL. At initial population, all constraints need to be defined, after that mutation and crossover operations are to be performed according to the defined constraints.

Sayyad et al. [11] applied metaheuristic search algorithms, including IBEA, NSGA and Strength Pareto Evolutionary Algorithm (SPEA), to achieve and compare the optimum results of SPL. Multi-objective optimization of FM to guide the developers of features selection for product derivation is important to satisfy the end user requirements under the given resources and constraints in FM of SPL. IBEA found much better optimum solutions compared to the other evolutionary algorithms with five objective functions. Efficiency comparison of algorithms is based on hyper volume, %correctness and spread parameters. IBEA aims to reduce the mutation and crossover operations by following the indicator user preference values.

Sayyad et al. [23] entertained the scalability problem of multi-objective optimization of FM to achieve the optimum solutions for product derivation of SPL by using IBEA. IBEA showed better performance in the context of hypervolume and correctness compared to NSGA-II. IBEA performs fully correct results in various large feature models from LVAT repository. IBEA suggested an indicator point for optimization that needs to achieve using different objective functions with crossover and mutation operations. However, NSGA-II compares the solutions and finally gives the minimum or maximum optimum points.

Olaechea et al. [29] addressed the problem of minimization and maximization of multi-objective optimizations such as lower costs and higher performance. The authors performed the comparison of exact and approximate optimum solutions on small and large FM. Findings of this study show the exact optimization is feasible on small FM however on large feature models approximate results are found. For exact optimization, Guided Improvement Algorithm (GIA) is feasible and for large feature models, IBEA performed better for approximate optimum solutions.

Xue et al. [30] applied IBEA for optimization of FM to minimize the cost in the context of increasing features and achieve optimum solutions with less constraint violations. The author proposed Differential Evaluation (DE) integrated

**TABLE 1. Comparison of Multi-objective optimization techniques.**

| Algorithm | Population | Operators | Domination Criteria | Achievements of Dominance Criteria | Number of Runs |
|---|---|---|---|---|---|
| NSGA-II [11], [23], [30] | One | Crossover, Mutation and Tournament selection | Calculate distance for each objective. More isolate product of these distance indicate fitness. | Dominant point indicates best optimal Solutions | Multiple (Different optimum solutions) |
| IBEA [11], [23], [31] | Main, archive | Crossover, Mutation and Environmental selection | Based on indicator value. Calculate minimum optimal value of each objective and these values are evaluated as indicator value. | User Objective values are favorable. | Multiple (Different optimum solutions) |
| MOO-BPNCC | All Configurations | Variabilities, Relationships, and Binary Patterns | Based on Binary values (0 and 1). Calculate the objective functions and find the value of each product. Perform comparison criteria in all product objective values and find the final optimum solution. | Dominated Mean value of all objective functions indicate the optimal solutions | One (Same Optimum solution in every run) |

with IBEA to minimize the execution time for large and complex feature models. The proposed approach is named as IBED; the combination of IBEA and DE. The optimum results indicate the best solutions with the consideration of cross-tree constraints.

Lian and Zhang [31] proposed the optimum solutions of non-functional and functional requirements of FM with Multi-Objective Evolutionary Algorithms (MOEA) with different parameters of IBEA, NSGA-II and SPEA2. The optimum results show the best performance of IBEA with less constraint violations and cross-tree constraints compared with other algorithms. IBEA performed optimization with indicator values from end users.

The approaches discussed above for optimization of FM clearly indicate the constraint violation of optimum solutions. For FM, fully correct optimum solutions are important due to selection or deselection of features for final products derivation. Therefore, in optimum solutions, if only one feature is selected that is not required in the actual product or have some cross-tree constraints with other features as well as cardinality constraints, the final products do not fulfill the end user requirements. Furthermore, the selection of features should be fully correct without any constraint violations and relationships. Our proposed algorithm for MOO-BPNCC works with binary patterns of features; value is 1 if the feature is part of the product and the value is 0 if the feature is not part of the final product.

## IV. FM MULTI-OBJECTIVE OPTIMIZATION

In current research, three MOEAs IBEA, NSGA-II and SPEA are primarily used for FM optimization of SPL. These MOEAs follow the basic operation of GA such as mutation and crossover for both single and multi-objective optimization to make new configuration. After crossover, every new configuration is compared with the previous dominated solution and if the solution dominates according to objective function, then it survives otherwise discard from solution set [32], [33].

Table 1 shows the comparison NSGA-II, IBEA and MOO-BPNCC. NSGA-II and IBEA perform optimization by using crossover and mutation operation to get optimum solutions. However, there is no criterion to find whether all possible solutions have been evaluated from objective functions or some solutions have missed during mutation and crossover operations. Furthermore, optimum solutions are not fully correct due to constraint violations.

FM constraint optimization consists on predefined constraints such as alternative, optional, OR group and cross-tree constraints. Crossover and mutation operations are performed on the basis of these constraints. Therefore, constraints need to be defined in every MEOAs for correct optimization. However, due to a large number of constraints and complex nested constraints, FM constraints violations occur and 100% correct solutions are not achievable. Optimization on E-shop FM, IBEA found 66.8% correct solutions with low parameters and 9.9% correctness with high parameter, NSGA performed 2.4% correctness with low parameters and 0.6% correctness with high parameters and SPEA performed 0.8% correctness with low parameters and 0.0% correctness with high parameters [11]. However, IBEA performed better than other optimization algorithms, but still not fully correct. Moreover, a single constraint violation in SPL configuration causes the final product derivation to fail. As shown in Fig. 1 alternative constraints are *inDays* and *unlimited*, only one can be selected. Therefore, if both *inDays* and *Unlimited* are selected, final SPL product fails.

## V. MOO-BPNCC OF FEATURE MODEL

BPNCC approach is used to obtain all possible unconstrained feature combinations. BPNCC algorithm solves all

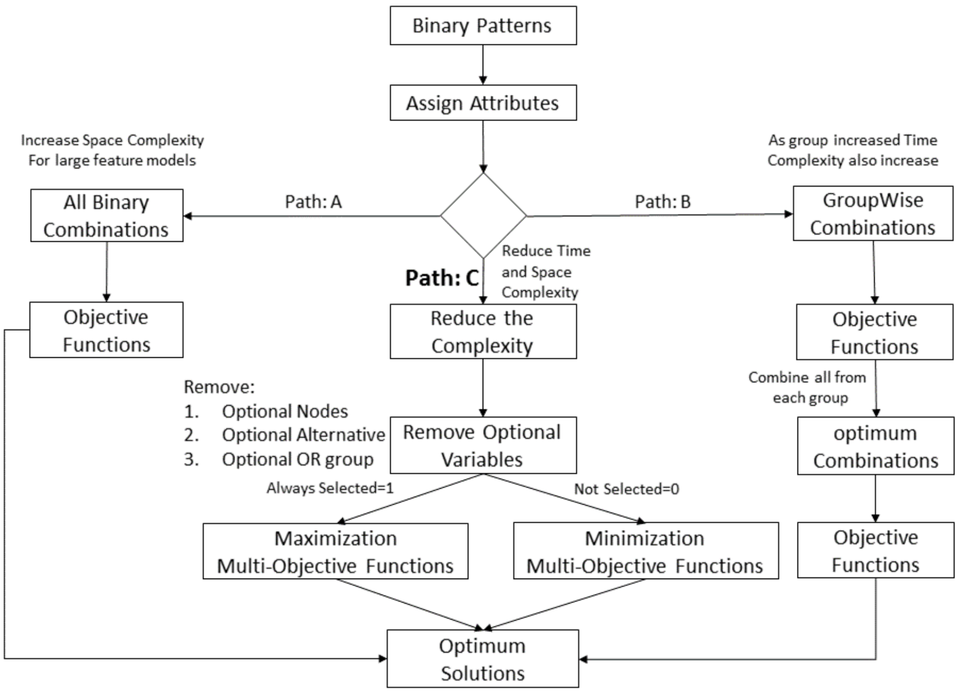


**FIGURE 2.** **Multi objective optimum binary pattern for nested cardinality constraints (MOO-BPNCC) process for optimum solutions.**

kind of constraints such as single level and nested constraints (alternative, optional and OR group) and final output is all products feature combinations in binary. In binary combinations, 1 indicates the selects and 0 indicates the non-selection of features in each product configuration. To find the binary combinations, BPNCC follow top-to-bottom approach with cardinality constraints and found all product configurations.

In this paper, we have proposed MOO-BPNCC to get the optimum solutions of SPL FM with a number of objective functions. The binary patterns enable the feature for selection or deselection in every configuration as 1 is used for selection and 0 deselections of a feature in final product derivation. Figure 2 shows MOO-BPNCC process to get optimum solutions. MOO-BPNCC starts with binary patterns of SPL configurations that are evaluated from BPNCC approach and assign the attribute values to terminal features with respect to objective functions as shown in Fig. 1. We have evaluated our proposed approach with four objective functions as given below:

- $Cost = \sum_{i=1}^{n} x_i$ where $n \epsilon Z$ and $x$ is the feature.
- $Performance = \sum_{i=1}^{n} x_i$ where $n \epsilon Z$ and $x$ is the feature.
- $CPU = \sum_{i=1}^{n} x_i$ where $n \epsilon Z$ and $x$ is the feature.
- $Memory = \sum_{i=1}^{n} x_i$ where $n \epsilon Z$ and $x$ is the feature.

We have adopted random attribute values for four objective functions as given in table 2.

For multi-objective optimum solutions (minimization or maximization) we have used the mean function as given in Eq. 1 to evaluate the objective function where all functions

**TABLE 2.** **Objective function attribute values.**

| Objective Functions | Attribute Values |
|---|---|
| Cost | 100−200 |
| Performance | 50-100 |
| CPU | 30-90 |
| Memory | 5-15 |

satisfy at one point.

$$\text{Mean Value for all Objectives} = \frac{1}{n} \sum_{i=1}^{n} f_i \tag{1}$$

In Eq. 1, f is function and n is the number of functions. Dominated, lower mean of minimized objective function of combinations, will survive and non-dominated combinations will be discarded. Finally, we have optimized feature combinations.

### A. MOO-BPNCC PATH A

Path A follows the complete data set of SPL product combinations that compute the objective functions one by one on each product combination for optimum solutions. This path is feasible on small feature models where less product combinations exist.

However, it is not feasible on large feature models where millions of product combinations exist. Due to lesser memory systems, it is not possible to compute objective functions simultaneously. Path A has space and execution time complexity with the increase of products. Algorithm 1 shows the process of path A.

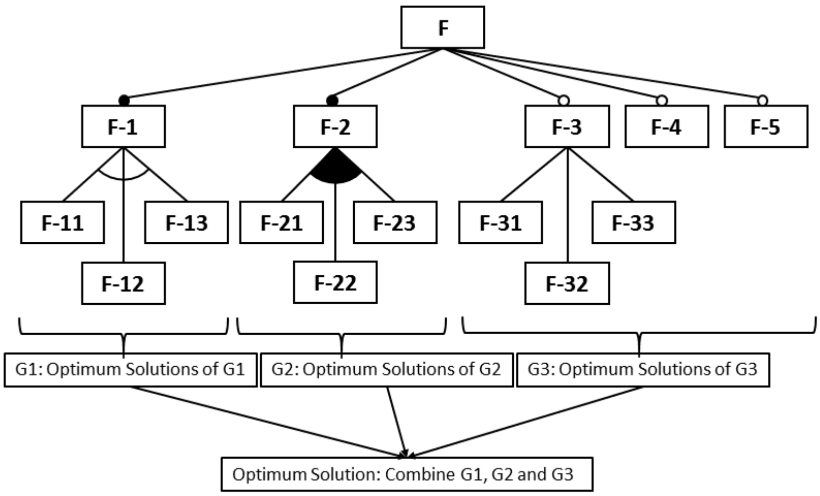


**FIGURE 3.** **GroupWise feature model (FM) optimal solutions.**

**Algorithm 1** MOO-BPNCC Path A

**Input** : Binary Features Combinations (BPNCC [24]).
t = Number of Configuration.
s = number of terminal features to be optimize.
x = number of objective functions.
**Output**: Minimized Optimum Features Combination.

```
for (i = 1 : t) do
    for (j = 1 : x)) do
        F(x) = ObjectiveFunctions;
        for (k = 1 : s) do
            Compute = attribute values for terminal
            features of jth combination;
        end
        a(i) = Compute Mean Value of x objective
        functions for each combination;
    end
    if (i > 1) then
        if (a(i − 1) > a(i)) then
            min = a(i);
        else
            min = a(i − 1);
        end
    end
end
```

However, this path is feasible for goal based optimum solutions by the end user requirements at any objective function points. To handle the space complexity, path B is a feasible approach to achieve the optimum solutions on less memory systems.

### B. MOO-BPNCC PATH B

We proposed path B to reduce the space complexity of large feature models where millions of product combinations exist. This path works on the basis of the GroupWise combination as shown in Fig. 3. The BPNCC approach computes the binary combinations of every group of FM and then combines all group combinations. By using path B, objective functions compute the optimum combinations from each group and then combine all group optimum combinations. Only dominated combinations from each group will survive and non-dominated combinations will be discarded. Objective functions need to apply one more time on final optimum combinations from each group to filter optimum solutions from group combinations.

**Algorithm 2** MOO-BPNCC Path B

**Input** : G = number of groups
**Output**: Minimized Optimum Features Combination.

```
for (i = 1 : G) do
    Define Relation of Gi with parent;
    Optional or Mandatory;
    if (Gi = a(i)) then
        Generate Binary Patterns;
    end
    Repeat;
    Enter number of Leaf Nodes;
    if (AllLeafNodes) then
        Generate Binary Patterns;
    else
        goto Repeat;
    end
    Recursive Call Path A (Evaluate Objective
    Functions);
    if (i > 1) then
        Combine G(i)and G(i − 1);
    end
end
```

Algorithm 2 shows the process of Path B, GroupWise optimum solutions, from each group and then combines all group optimum combinations.

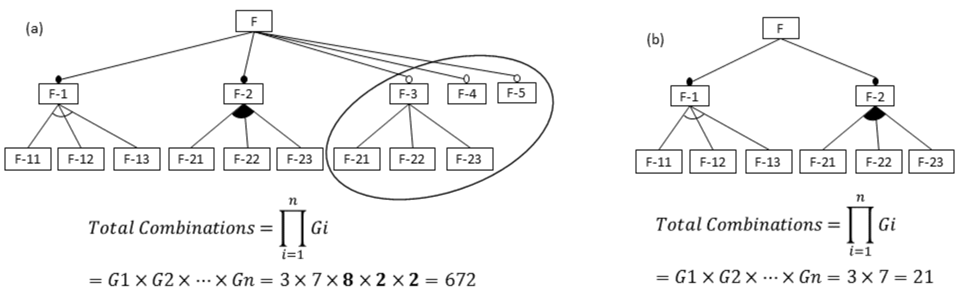


**FIGURE 4.** Feature model configurations (a) including optional variables (b) excluding optional variables.

This path is feasible on large and small FM due to Group-Wise optimum solutions. By using this path, space complexity can be reduced, but time complexity will increase as computation of objective functions applied two times on each group separately and on final dominated combinations from every group. Furthermore, objective functions. Therefore, this path is feasible to reduce the space complexity but infeasible for computation time perspectives.

### C. MOO-BPNCC PATH C

Path C is most suitable to achieve optimum solutions from large and small feature models with less execution time and reduce space complexity. For optimum solutions, optional variables are always not selected; 0 (attribute values is 0) for minimization of objective functions and always selected, 1 (attribute values is 1) for maximization of objective functions. Therefore, optional variable features can be during all product combinations. Three types of optional variables exist in FM given below:

**Algorithm 3** MOO-BPNCC Path C

**Input** : G = number of groups
**Output**: Minimized Optimum Features Combination.

```
for (i = 1 : G) do
    Define Relation of Gi with parent;
    Optional or Mandatory;
    if (Gi = Optional||LeafNodes) then
        Do Nothing;
    else
        Repeat;
        Number of Children;
        if (All LeafNodes) then
            Generate Binary Patterns;
        else
            Goto Repeat;
        end
    end
    Recursive Call Path A (Evaluate Objective
    Functions);
end
```

- Optional Leaf Nodes
- Optional Alternative Group
- Optional OR Group

In Fig. 4, values of optional variables always 0 for minimization of the objective function and 1 for maximization of objective functions. Moreover, optional variables have a significant role on the complexity of FM, from 627 binary combinations only 21 remaining combinations are available to optimize. Therefore, FM complexity can be reduced by removing of these features during optimum solutions.

**TABLE 3.** Values of optional variables.

| Objective Functions | F-21 | F-22 | F-23 | F-4 | F-5 |
|---|---|---|---|---|---|
| Minimization | 0 | 0 | 0 | 0 | 0 |
| Maximization | 1 | 1 | 1 | 1 | 1 |

Table 3 shows the values of option variables for minimized and maximized an objective function. All optional variables are not part of final product derivation of minimized optimal solutions and are always part of final product derivation of maximized optimal solutions.

Algorithm 3 shows the process of Path C; the optimum combinations found by excluding the optional variables.

## VI. EXPERIMENT AND PERFORMANCE EVALUATION

We have reduced the complexity of Fig. 1 IoT-based application FM by using Path C. Figure 5 shows the comparison of Cost, Fig. 6 shows the comparison of performance, Fig. 7 shows the comparison of CPU and Fig. 8 shows the comparison of memory by using Path A (a), Path B and Path C (b) that clearly indicate the same value of first fifteen products. From one to fifteen products configurations, the sum of attribute values of cost and performance is same due to removal of the three optional features. Therefore, the minimum optimum value is lies at first fifteen combinations.

We used MATLAB R2015b tool and system specifications 6GB RAM, Intel(R) Core(TM) i3 with 3.30GHz processor for experimental verification of MOO-BPNCC. To verify

**TABLE 4. Comparison of space and time complexity of Path A, B and C.**

| Feature Model | #Features | Optional Terminal Features | MOO-BPNCC Paths | #Combinations | Mean Time 20 Runs | Constraints Violations |
|---|---|---|---|---|---|---|
| Security Settings | 18 | 3 | Path A | 480 | 0.008 | 0 |
| | | | Path B | #Groups=5 | O(#Groups) | 0 |
| | | | Path C | 60 | 0.006 | 0 |
| Speech Recognition | 32 | 12 | Path A | 98304 | 0.510 | 0 |
| | | | Path B | #Groups=9 | O(#Groups) | 0 |
| | | | Path C | 24 | 0.004 | 0 |
| BerkeleyDB | 29 | 17 | Path A | 1354752 | 12.403 | 0 |
| | | | Path B | #Groups=10 | O(#Groups) | 0 |
| | | | Path C | 2646 | 0.015 | 0 |

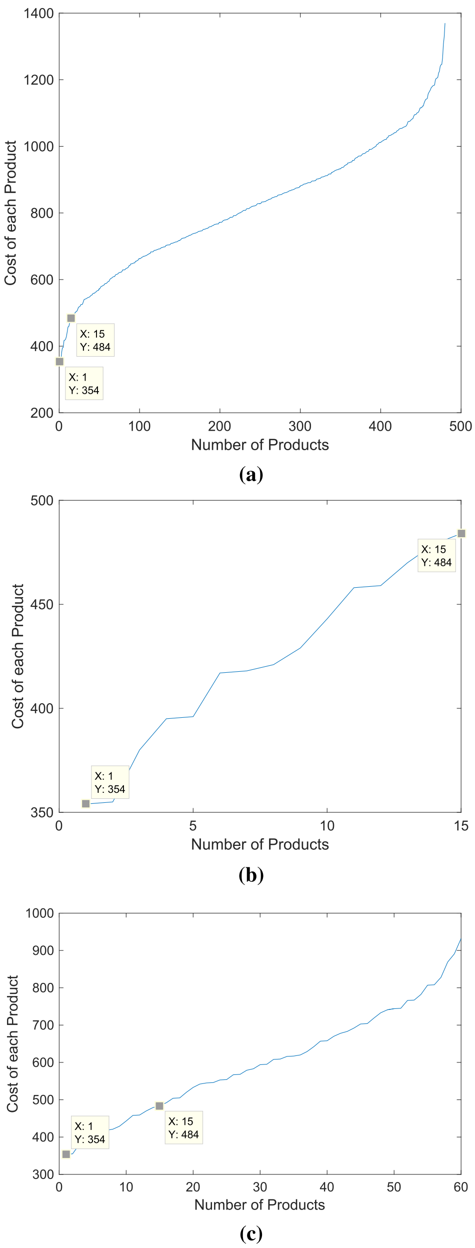


**FIGURE 5. Cost objective function for (a) Path A, (b) Path B and (c) Path C.**

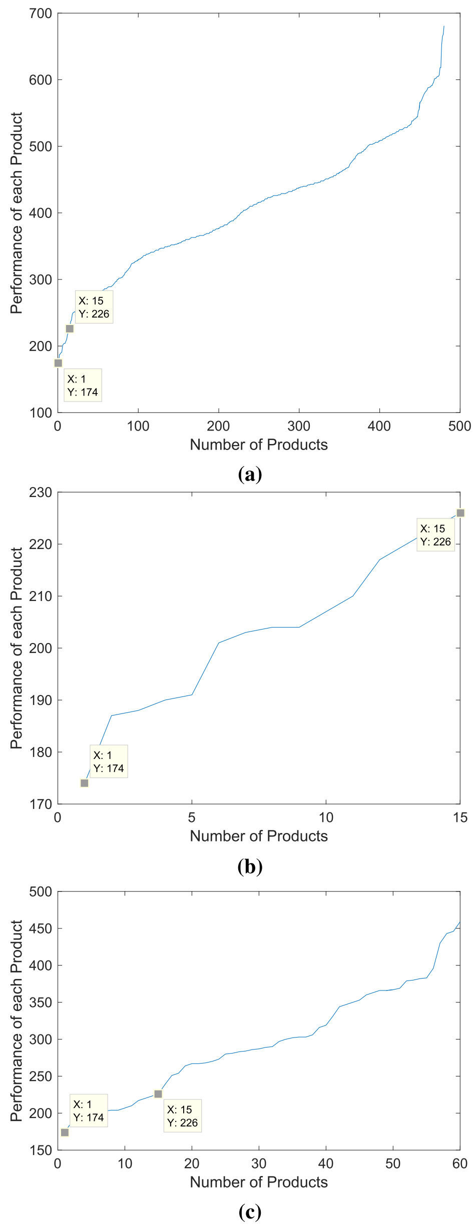


**FIGURE 6. Performance objective function for (a) Path A, (b) Path B and (c) Path C.**

MOO-BPNCC, we applied on small and large feature models from SPLOT [34] and calculate time and space complexity as shown in table 4.

By using path A, all configurations need to be optimized that increase space and execution time with no constraint violations (i.e., correct optimum solutions). By using Path B,

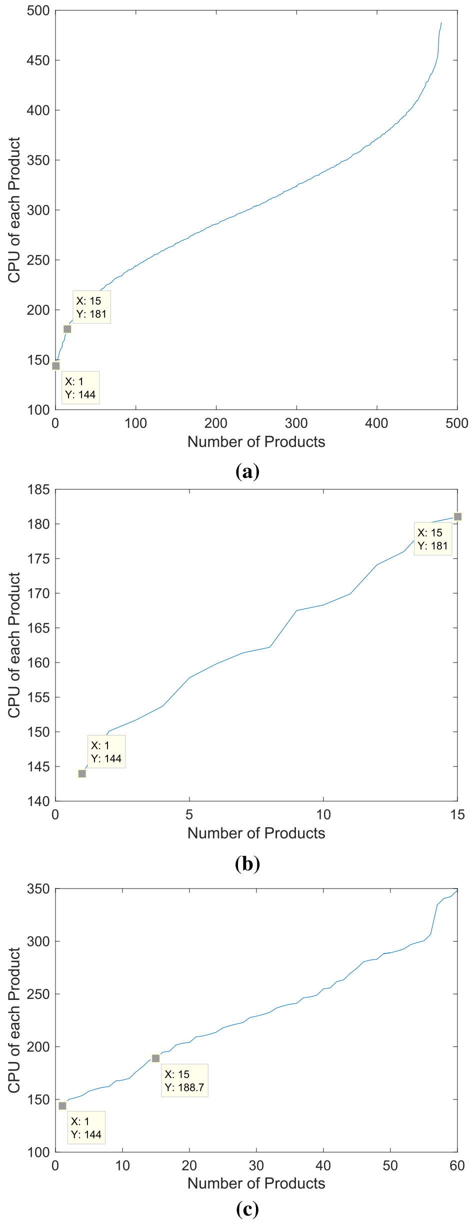


**FIGURE 7.** CPU objective function for (a) Path A, (b) Path B and (c) Path C.

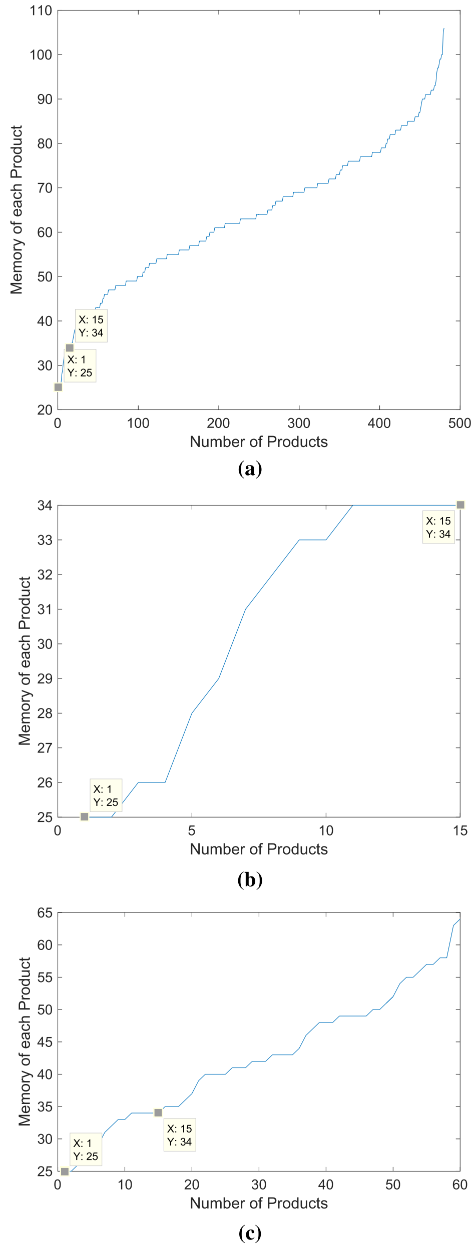


**FIGURE 8.** Memory objective function for (a) Path A, (b) Path B and (c) Path C.

as the number of groups increase, time complexity also increase. Path C is the most effective and efficient with less configurations; less space and less execution time with zero constraint violation.

Minimized optimum solutions by using Path A, Path B and Path C is shown Fig. 9. Having the same minimized optimum solutions of Path A (with optional features), Path B (GroupWise evaluation) and Path C (without optional variables) indicate that optional variables are not necessary for minimized and maximized optimum solutions. For maximized optimum solutions, all attribute values of optional features need to be calculated according to objective functions by using Path C.

## VII. DISCUSSION AND LIMITATIONS OF MOO-BPNCC

In this study, we have proposed MOO-BPNCC to achieve the minimized and maximized optimum solutions of the contextual variability of IoT-based FM. From experimental results, we have observed that from three MOO-BPNCC paths, path C is more efficient to obtain optimum solutions. However, our proposed approach does not cover the goal-based optimum solutions (i.e., reference point base). Moreover, in our experimental work, we only considered the basic predefined relationships constraints of FM, but do not consider the cross-tree constraints. Furthermore, we have applied the

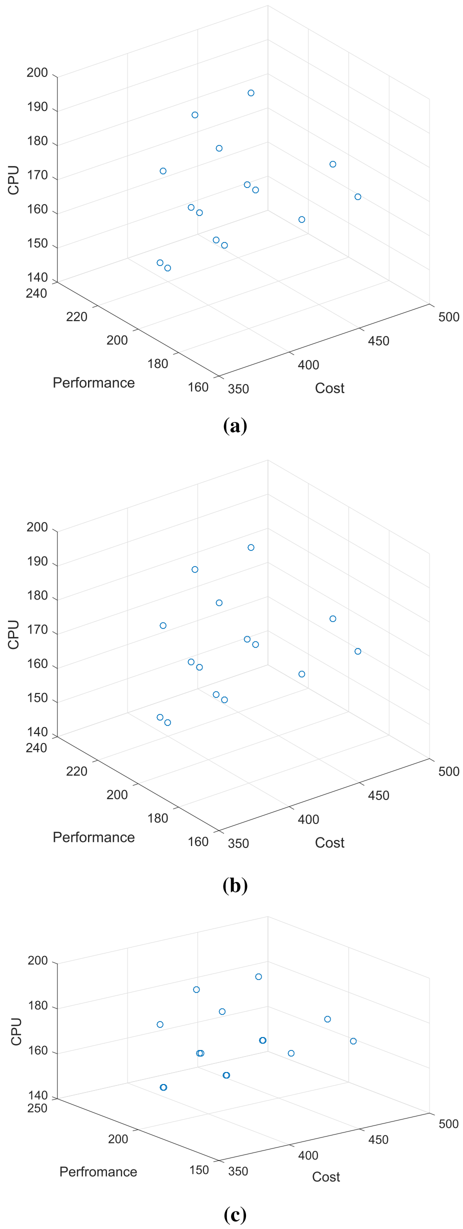


**FIGURE 9.** **Minimized multi-objective optimum solutions (a) Path A, (b) Path B and (c) Path C.**

proposed approach for minimized objective functions and found correct optimum solutions with less space and time complexity by using Path C. In our experimental work, maximized objective functions do not exist. However, from given process model, MOO-BPNCC is effective and efficient for both minimized and maximized objective functions.

## VIII. CONCLUSION

Software Product Line is extensively used in industry for quick development with less cost and time to market by applying the reusability of existing resources. FM is used to manage the contextual variables and common features of SPL. Due to the existence of contextual variability in IoT applications, it is important to manage and increase the reusability of IoT application features for quick development and time to market with less cost. Optimization is the best paradigm to handle the contextual variability according to end user requirements. In this paper, we extended our previous proposed approach BPNCC to MOO-BPNCC to get multi-objective optimum solutions. We proposed three paths of MOO-BPNCC and presented limitations of each path to get optimum solutions. However, Path C is more feasible in case of less execution time and space with reducing the complexity of features combinations by ignoring the optional variables from FM during the optimum process. Furthermore, our experimental results show, path C is the best process to get optimum features combinations for product derivations.

In future work, we will enhance the optimum solutions with cross-tree constraints and goal-base optimization. Furthermore, we will perform multi-objective optimization with the priority of specific function.

## REFERENCES

[1] S. U.-J. Lee, "An effective methodology with automated product configuration for software product line development," *Math. Problems Eng.*, vol. 2015, Jun. 2015, Art. no. 435316.
[2] K. Lee, K. C. Kang, M. Kim, and S. Park, "Combining feature-oriented analysis and aspect-oriented programming for product line asset development," in *Proc. 10th Int. Softw. Product Line Conf.*, 2006, pp. 1–10.
[3] M. Sinnema and S. Deelstra, "Classifying variability modeling techniques," *Inf. Softw. Technol.*, vol. 49, no. 7, pp. 717–739, 2007.
[4] M. Körner, S. Herold, and A. Rausch, "Composition of applications based on software product lines using architecture fragments and component sets," in *Proc. WICSA Companion Volume*, 2014, pp. 1–13.
[5] A. Abbas, I. F. Siddiqui, and S. U.-J. Lee, "Goal-based modeling for requirement traceability of software product line," *J. Theor. Appl. Inf. Technol.*, vol. 94, no. 2, p. 327, 2016.
[6] A. Abbas, Z. Wu, I. F. Siddiqui, and S. U.-J. Lee, "An approach for optimized feature selection in software product lines using union-find and genetic algorithms," *Indian J. Sci. Technol.*, vol. 9, no. 17, pp. 1–8, 2016.
[7] L. Zheng, C. Zhang, Z. Wu, and M. Liu, "Managing resource repository of a software product line with feature model," in *Proc. Int. Conf. Comput. Intell. Softw. Eng. (CiSE)*, Dec. 2009, pp. 1–4.
[8] J. Lee and K. C. Kang, "Feature binding analysis for product line component development," in *Proc. Int. Workshop Softw. Product-Family Eng.*, 2003, pp. 250–260.
[9] J. Gillain, S. Faulkner, P. Heymans, I. Jureta, and M. Snoeck, "Product portfolio scope optimization based on features and goals," in *Proc. 16th Int. Softw. Product Line Conf.*, vol. 1. 2012, pp. 161–170.
[10] A. Abbas, I. F. Siddiqui, and S. U.-J. Lee, "Multi-objective optimization of feature model in software product line: Perspectives and challenges," *Indian J. Sci. Technol.*, vol. 9, no. 45, pp. 1–7, 2016.
[11] A. S. Sayyad, J. Ingram, T. Menzies, and H. Ammar, "Optimum feature selection in software product lines: Let your model and values guide your search," in *Proc. 1st Int. Workshop Combining Modelling Search-Based Softw. Eng.*, 2013, pp. 22–27.
[12] N. Jabeur, A. U.-H. Yasar, E. Shakshuki, and H. Haddad, "Toward a bio-inspired adaptive spatial clustering approach for IoT applications," *Future Generat. Comput. Syst.*, pp. 1–9, May 2017.
[13] S. N. Han and N. Crespi, "Semantic service provisioning for smart objects: Integrating IoT applications into the Web," *Future Generat. Comput. Syst.*, vol. 76, pp. 108–197, Nov. 2017.
[14] I. F. Siddiqui, S. U.-J. Lee, A. Abbas, and A. K. Bashir, "Optimizing lifespan and energy consumption by smart meters in green-cloud-based smart grids," *IEEE Access*, vol. 5, pp. 20934–20945, 2017.
[15] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[16] C. M. de Farias *et al.*, "COMFIT: A development environment for the Internet of Things," *Future Generat. Comput. Syst.*, vol. 75, pp. 128–144, Oct. 2017.
[17] M. Endler, J.-P. Briot, V. P. Almeida, F. S. E. Silva, and E. Haeusler, "Towards stream-based reasoning and machine learning for IoT applications," in *Proc. IEEE Intell. Syst. Conf. (IntelliSys)*, Sep. 2017, pp. 1–11.
[18] A. Abbas, I. F. Siddiqui, and S. U.-J. Lee, "Contextual variability management of IoT application with xml-based feature modelling," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 6, pp. 1300–1308, 2017.
[19] I. Ayala, M. Amor, L. Fuentes, and J. M. Troya, "A software product line process to develop agents for the IoT," *Sensors*, vol. 15, no. 7, pp. 15640–15660, 2015.
[20] P. D. A. dos Santos Neto, R. Britto, R. de Andrade Lira Rabêlo, J. J. de Almeida Cruz, and W. A. L. Lira, "A hybrid approach to suggest software product line portfolios," *Appl. Soft Comput.*, vol. 49, pp. 1243–1255, Dec. 2016.
[21] G. A. E.-N. A. Said, A. M. Mahmoud, and E.-S. M. El-Horbaty. (2014). "A comparative study of meta-heuristic algorithms for solving quadratic assignment problem." [Online]. Available: https://arxiv.org/abs/1407.4863
[22] Y.-L. Wang and J.-W. Pang, "Ant colony optimization for feature selection in software product lines," *J. Shanghai Jiaotong Univ. (Sci.)*, vol. 19, no. 1, pp. 50–58, 2014.
[23] A. S. Sayyad, J. Ingram, T. Menzies, and H. Ammar, "Scalable product line configuration: A straw to break the camel's back," in *Proc. IEEE/ACM 28th Int. Conf. Autom. Softw. Eng. (ASE)*, Nov. 2013, pp. 465–474.
[24] A. Abbas, I. F. Siddiqui, S. U.-J. Lee, and A. K. Bashir, "Binary pattern for nested cardinality constraints for software product line of IoT-based feature models," *IEEE Access*, vol. 5, pp. 3971–3980, 2017.
[25] T. Thüm, C. Kästner, F. Benduhn, J. Meinicke, G. Saake, and T. Leich, "FeatureIDE: An extensible framework for feature-oriented software development," *Sci. Comput. Programm.*, vol. 79, pp. 70–85, Jan. 2014.
[26] A. S. Karataş and H. Oğuztüzün, "Attribute-based variability in feature models," *Requirements Eng.*, vol. 21, no. 2, pp. 185–208, 2016.
[27] F. Loesch and E. Ploedereder, "Optimization of variability in software product lines," in *Proc. 11th Int. Softw. Product Line Conf. (SPLC)*, Sep. 2007, pp. 151–162.
[28] J. Guo, J. White, G. Wang, J. Li, and Y. Wang, "A genetic algorithm for optimized feature selection with resource constraints in software product lines," *J. Syst. Softw.*, vol. 84, no. 12, pp. 2208–2221, 2011.
[29] R. Olaechea, D. Rayside, J. Guo, and K. Czarnecki, "Comparison of exact and approximate multi-objective optimization for software product lines," in *Proc. 18th Int. Softw. Product Line Conf.*, vol. 1. 2014, pp. 92–101.
[30] Y. Xue *et al.*, "IBED: Combining IBEA and DE for optimal feature selection in software product line engineering," *Appl. Soft Comput.*, vol. 49, pp. 1215–1231, Dec. 2016.
[31] X. Lian and L. Zhang, "Optimized feature selection towards functional and non-functional requirements in software product lines," in *Proc. IEEE 22nd Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Mar. 2015, pp. 191–200.
[32] C. Henard, M. Papadakis, M. Harman, and Y. Le Traon, "Combining multi-objective search and constraint solving for configuring large software product lines," in *Proc. IEEE/ACM 37th IEEE Int. Conf. Softw. Eng. (ICSE)*, vol. 1. May 2015, pp. 517–528.
[33] A. S. Sayyad, T. Menzies, and H. Ammar, "On the value of user preferences in search-based software engineering: A case study in software product lines," in *Proc. Int. Conf. Softw. Eng.*, May 2013, pp. 492–501.
[34] *SPLOT—Software Product Line Online Tools*. Accessed: Jun. 2, 2017. [Online]. Available: http://ec2-52-32-1-180.us-west-2.compute.amazonaws.com:8080/SPLOT/index.html.

**ASAD ABBAS** (S'17) received the B.S. degree in information technology from the University of the Punjab, Pakistan, in 2011. He is currently pursuing the M.S. degree, leading to PhD Program, with the Department of Computer Science and Engineering, Hanyang University ERICA Campus, Ansan, South Korea, funded by the Higher Education Commission of Pakistan in 2014. His research interests include software product line, software requirement traceability, and IoT applications.

**ISMA FARAH SIDDIQUI** received the B.E. degree in software engineering and the M.E. degree in information technology from the Mehran University of Engineering and Technology, Pakistan, in 2006 and 2008, respectively, and the Ph.D. degree from Hanyang University ERICA Campus, South Korea, in 2008, funded by the Higher Education Commission, Pakistan. Since 2006, she has been with the Department of Software Engineering, Mehran UET, where she is currently serving as an Assistant Professor. Her research interests include machine learning and data analytics in areas of smart environment, semantic Web, IoT, green cloud, and big data.

**SCOTT UK–JIN LEE** received the B.E degree in software engineering and the Ph.D. degree in computer science from The University of Auckland, New Zealand. After the Ph.D. degree, he was a Post-Doctoral Research Fellow at the Commissariat à l'énergieatomique et aux énergies alternatives, France. He has been with the Department of Computer Science and Engineering, Hanyang University ERICA Campus, South Korea, since 2011. He is currently serves as the Head of the Department of Software for Emerging Technology major. His research interests include software engineering, formal methods, Web, and IoT. He is also a member of the Korean Institute of Information Scientists and Engineers and the Korean Society of Computer and Information. He has served as an editor, the technical chair, and a committee member for several journals and conferences.

**ALI KASHIF BASHIR** (M'15–SM'16) received the Ph.D. degree in computer science and engineering from Korea University, South Korea. He held appointments with Osaka University, Japan; Nara National College of Technology, Japan; the National Fusion Research Institute, South Korea; Southern Power Company Ltd., South Korea, and the Seoul Metropolitan Government, South Korea. He is currently an Associate Professor with the Faculty of Science and Technology, University of the Faroe Islands, Faroe Islands. He is also attached to the Advanced Network Architecture Lab as a Joint Researcher. He is supervising/co-supervising several graduate (M.S. and Ph.D.) students. His research interests include cloud computing, NFV/SDN, network virtualization, network security, IoT, computer networks, RFID, sensor networks, wireless networks, and distributed computing. He is actively involved in organizing workshops and conferences. He has chaired several conference sessions, gave several invited and keynote talks, and reviewed the technology leading articles for journals, such as the IEEE Transactions on Industrial Informatics, the *IEEE Communication Magazine*, the IEEE Communication Letters, the IEEE Internet of Things, and the IEICE Journals, and conferences, such as the IEEE INFOCOM, the IEEE ICC, the IEEE GLOBECOM, and the IEEE Cloud of Things. He is serving as the Editor-in-chief for the IEEE Internet Technology Policy Newsletter and the IEEE Future Directions Newsletter. He is an Editorial Board Member of journals, such as the IEEE Access, the *Journal of Sensor Networks*, and the *Data Communications*. He has also served/serving as a Guest Editor on several special issues in journals of the IEEE, Elsevier, and Springer.

**WALEED EJAZ** (S'12–M'14–SM'16) received the B.Sc. degree in computer engineering from the University of Engineering and Technology at Taxila, Pakistan, the M.Sc. degree in computer engineering from the National University of Sciences and Technology, Islamabad, Pakistan, and the Ph.D. degree in information and communication engineering from Sejong University, Seoul, South Korea, in 2014. He has completed certificate courses on teaching and learning in higher education from the Chang School, Ryerson University. He held academic and research positions at Ryerson University, Toronto, ON, Canada; Carleton University, Ottawa, ON, Canada; and Queen's University, Kingston, ON, Canada. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Sultan Qaboos University, Muscat, Oman. His current research interests include Internet of Things, energy harvesting, 5G cellular networks, and mobile cloud computing. He is handling several special issues in prestigious journals, as well as in organizing committees of several major IEEE conferences and workshops. He is a registered Professional Engineer in the Province of Ontario, Canada. He is an Associate Editor of the *IEEE Communications Magazine*, the IEEE Canadian Journal of Electrical and Computer Engineering, and the IEEE Access.

**NAWAB MUHAMMAD FASEEH QURESHI** received the B.E. degree in software engineering and the M.E. degree in information technology from the Mehran University of Engineering and Technology, Pakistan, respectively, and the Ph.D. degree in computer engineering from Sungkyunkwan University, South Korea. He is currently an Assistant Professor with Sungkyunkwan University, South Korea, where, he is actively involved in the Big Data Project and discusses new trends in IoT-enabled data analytics. His research interests include big data analytics, context-aware data processing of the Internet of Things, and cloud computing. He has been an active TPC member of various international conferences, including CSA2017, WCSN2017, and IWWCN2017. He received the SAMSUNG scholarship for his Ph.D. degree and the Superior Research Award from the College of Information and Communication Engineering on account of his research contributions and performance during Ph.D. studies. He is a Reviewer of various renowned journals, such as the IEEE Transactions on Industrial Informatics, *Future Generation Computer Systems*, the IEEE Access, *The Journal of Supercomputing*, *Wireless Personal Communication*, and *KSII Transactions on Internet and Information Systems*.