*I.J. Information Technology and Computer Science*, **2015, 09, 31-37**
Published Online August 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.09.05

# Multi Objective Optimization Model using Preemptive Goal Programming for Software Component Selection

**Jagdeep Kaur**
CSE Department, ITMU, Gurgaon, Haryana, India
Email: jagdeep_kaur82@rediffmail.com

**Pradeep Tomar**
School of Information and Communication Technology, Gautam Budhha University, Greater Noida, U.P., INDIA
Email: parry.tomar@gmail.com

***Abstract*****—**To achieve successful reusability of components a disciplined development approach is required which is the component based software engineering(CBSE).The software component selection is a vital part of this approach. It consists of defining an evaluation criteria based on user requirements and depending on this the repository is searched and shortlisted components are presented to the user. Due to availability of large number of components offering same type of functionality it is difficult to select a particular component based on available description. This paper presents a multiobjective optimization model for component selection purpose and solves it using preemptive goal programming approach by using an optimization tool LINDO. Subsequently, an illustrative case study is given where the components are taken from an online repository and goal programming is applied for getting the most optimal component. However, this model is applicable when the repository is small but for larger set of components it needs to be validated.

***Index Terms*****—**Multiobjective Optimization, Goal Programming, Component Selection, Hard Constraint, Goal Constraint.

## I. INTRODUCTION

The software can be defined as the set of programs along with associated configuration data and documentation. Initially, the traditional software development paradigms were used to develop complex and robust software. But with the demand of lower software production and maintenance costs, quick delivery of systems and increased software quality the reuse based development has geared up. Apart from cost reduction the reusability offers many advantages like increased trustworthiness due to tried and tested software, reduction in process risks, standard compliance and accelerated delivery time. Different approaches are being used to implement reusability viz. Design Patterns, Program Libraries, Legacy system wrapping, Application frameworks, Service-Oriented Systems, Program generators, Aspect-oriented Software Development and Component- Based Software Development[1]. The Component Based Software Development (CBSD) is a paradigm that is used for creating complex software system from the existing software components. The software component is described in various ways by the experts in the literature. One definition given by *Brown* [2] says that component is an independently deliverable piece of functionality providing access to its services through interfaces. A software component as defined by *Szyperski* in[3] is “A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties”. The major advantage of CBSD is the reusability of the functionality across various applications that results in a software of higher quality and better reliability. Software practitioners, researchers and academicians are trying to improve the software development practices by improving the design methodology, using different notations for representing system’s functionality and by promoting reusability using COTS.

The software are developed using the components from various repositories. While doing component selection the quality should be maintained and for this many approaches are used. One of the approach is to maintain the quality throughout the lifecycle model in CBSD using a modified V cycle based on reliability [4]. Another way to assess the quality is by use of software metrics based on information collected during runtime[5].In the repository the description about these components are provided by the developers. The components are selected mainly based on its functionality from the repository but the non- functional properties or the information provided by the developers about the components also play a crucial role in component selection. As the number of available components in the repository grows, the selection of set of components based on a set of functional requirements and on the other hand minimizing or maximizing other objectives like price, number of components etc has become extremely difficult. The software component selection problem can be treated as as Multi-Criteria Decision Making (MCDM) or Multi-Objective Optimization(MOO) .

MCDM or MOO problems are those where decisions are to be made over the set of available choices by having multiple usually conflicting attributes [6].The main aim of MCDM is to assist the decision maker in making the best choice by finding out the best alternative or by ranking the alternatives based on the choices. [7] Hence, the software component selection can be considered as the Multi-Criteria Decision Making Problem. In Multicriteria Optimization deals with simultaneously optimizing two or more conflicting objectives subject to certain constraints. When the criteria are different there are different methods to tackle with these problems like by using weighted sum method or by assigning priorities to the criteria. However, the weight coefficients in the former approach need to be determined explicitly which always depend on the subjective judgment of the developers. Whereas, it is easy to assign priorities on the objective by the developers.

There are different ways to solve these types of problems according to the conditions namely No preference method where the multiobjective optimization problem is solved using a simple method without consulting the decision maker. In Apriori methods the decision maker sets preferences before applying optimization. In case of Posteriori method representative sets of Pareto optimal solutions are presented to the decision maker so that s(he) can choose the best among them. As far as interactive methods are concerned it allows the decision maker to guide the search by alternating optimization and preference articulation iteratively. In case of Weighted sum method, a set of objectives are converted into single objective by pre-multiplying each user objective by user given weights. Similarly, €- constraint method takes one objective and keep the other objectives within the user specified values. Apart from these other methods are Weighted Metric Method, Benson Method, Value Function etc. Another important method is Goal Programming where instead of trying to optimize all the objectives we set goal for the objective values and try to meet these goals instead. Here, the objectives are assigned a specific value and a solution is found that minimizes the weighted sum of deviations of these objective functions from their respective goals. Depending upon types of goals there are two categories of goal programming “Pre-emptive Goal Programming” and “No- Preemptive Goal Programming”. In former, the priorities of the goals are in hierarchical order whereas in latter all the goals are of comparable importance. Goal Programming can be applied to different mathematical models like Linear programming, Non-Linear programming, Integer Programming, Zero-One Goal Programming etc .Our proposed model will make use of Zero- One preemptive Goal Programming as the software components are either selected or rejected.

This paper is organized into five sections. The related work in discussed in section 2, the next section 3 gives detailed discussion of proposed solution to multiobjective optimization model using preemptive goal programming. The section 4 gives an illustrative case study using LINDO taken from an online repository and section 5 is for results and discussions. The conclusion of the paper is given in section 6.

## II. RELATED WORK

So far, many researchers have proposed numerous approaches for software component selection as multiobjective optimization. Initially, the article by Vescan [8] proposes the component selection as multiobjective optimization with the problem of selecting components from the available set while minimizing the number of components selected and the cost of the components.

In the Multiobjective Optimization or Multicriteria decision making the earlier work done by [9] considers minimizing the number of used components, number of new requirements, number of provided interfaces and number of initial requirements that are not in the solution. It is solved using evolutionary algorithm which proceeds from the solutions found so far.

Another Multiobjective Optimization for software component selection under multi application development at a time was proposed in [10]. The objective is to minimize the total procurement cost and total adaptation cost of the selected components considering reusability and compatibility simultaneously. It is solved by using customary genetic algorithm which may not always give optimal result.

Two Multiobjective Models were proposed by Jha et al. in [11], in the first one the objective is to maximize the reliability and minimize the overall cost with the multiple constraints like either the component can be build or purchased, redundancy constraints, delivery time constraints & probability of failure free in house component to be delivered. In the second optimization model, additional constraints are introduced to represent the compatibility among the alternatives of the components, but the objective is the same.

According to [12] the objective function of the optimization model consists of software functionality and software quality like the reliability and user satisfaction. The user satisfaction is measured based on a set of factors. The model is solved using Binary particle swarm optimization.

Earlier some work was done for COTS Component Selection by using Goal Programming approach. As highlighted in [13] in order to choose components for a fault tolerant modular system such that the reliability of the system is increased while the overall price is reduced. Under this situation the use of chance constrained goal programming is suggested as it minimizes the nonconformity between the attainment level of the objectives and the goals set for them.

Another work by [14] considers the selection problem of repairable component for parallel series system as a multiobjective optimization problem. Two models are proposed and solved using preemptive goal programming. The solution improves the reliability of the selected system with compromised solution of repairable changes.

After studying the previous work it was noticed that all the techniques were considering cost to be reduced while taking care of different factors like non functional requirements, compatibility, buy versus build decision etc. But in case components are retrieved from online repository as a result of keyword search, it is difficult to retrieve the best suited component according to the given parameters like Bestseller based rating, Review based rating, download based rating etc. For these types of scenarios the authors propose a new multiobjective optimization model that is solved using preemptive goal programming.

## III. Proposed Solution To Multiobjective Optimization Model Using Preemptive Goal Programming

In Multi objective optimization the goal programming method is one of the oldest technique which works on the principle of minimizing the deviation of each of the objective from the desired level. An important review was done by Orumie in [15] and comparison of different algorithms was made on the basis of computational time and accuracy. However, in the software component selection technique the application developer needs to select the desired component from the repository based on his/her requirements and the goal programming method has been rightly adapted.

The Preemptive Goal Programming is considered for solving this optimization problem as the idea behind this is that lower priority goals should not be achieved at the cost of higher priority goals, they are preempted. In the Goal Programming method the objectives are converted into goals and for each goal a pair of deviation variables are defined. Mathematically, it is represented as:

$$G_i : f_i(x) + U_i - E_i \text{ for } i=1,2,3\ldots.N$$

where N is the number of Goals

$f_i(x)$ is the mathematical expression for the goal

$U_i$ & $E_i$ are the deviational variables for each goal.

$U_i$ the amount by which the left side falls short of (under) its right hand side value

$E_i$ is the amount by which the left side exceeds its right hand side value.

After formulating the goals the next target of Goal Programming is to minimize the Detrimental variables. There are three different cases as highlighted in [16] :

i). If $f_i(x) >= B_i$ , Goal is attached to maximization type of objective. Here, the decision maker does not want under achievement with respect to target $B_i$. Hence, $U_i$ is to be minimized.

ii). ii) If $f_i(x) =< B_i$ , Goal is attached to minimization type of objective. Here, the decision maker does not want over achievement with respect to target $B_i$. Hence, $E_i$ is to be minimized.

iii). If $f_i(x) = B_i$ , Goal is to be achieved exactly. Here, the decision maker neither wants under achievement nor over achievement with respect to target $B_i$. Hence, $(U_i + E_i)$ is to be minimized as both are equally unwanted.

This model is applicable to any situation where the developer has clear idea in his/her mind about the requirements, constraints and the priorities of these requirements. In terms of goal programming, the hard constraints, goal constraints and the priorities need to be defined. The number of hard constraints and goal constraints can be more than one. Even the priorities can range from two to some greater number. In the present case, the hard constraints are the budget and the number of components to be retrieved. The goal constraint according to the priorities are bestseller rating is at the first priority, secondly download rating and thirdly review based rating is considered. Moreover, the component is either selected or rejected so Zero-One goal programming is used.

The general Zero-One Goal Programming model in selecting the optimal set of components can be stated as follows:

$$\text{Minimize } Z = P_i\,(w_n\, p_1,\, w_m\, p_2,\, w_o\, p_3) \qquad (1)$$

Subject to

Hard Constraints

$$\sum_{j=1}^{20} Pj\ Cj\ < = \text{Budget}_{\text{Max\_allocated}} \qquad (2)$$

That means the total budget of the selected components should be within the budget limitations

$$\sum_{j=1}^{20} Cj = \text{No.\_of\_Components\_to\_be\_selected} \qquad (3)$$

Goal Constraints

$$b_i C_i + b_j C_j^i + n_j - p_j \leq \text{sum\_of\_top\_bestseller\_rating} \qquad (4)$$

$$d_j C_j + d_j C_j^i + n_j - p_j \leq \text{sum\_of\_top\_downloaded\_rating} \qquad (5)$$

$$r_j C_j + r_j C_j^i + n_j - p_j \leq \text{sum\_of\_top\_reviewbased\_rating} \qquad (6)$$

$$C_j = 0 \text{ or } 1 \qquad (7)$$

$$c_j n_j p_j \geq 0$$

where

$P_i$ = Sum i preemptive priority($P_1 > P_2 > P_3$) for i= 1,2,3…n

$w_n$ = The weightage assigned to the 20 components

$n_j, p_j$ = The negative and the positive deviation variables are for i =1,2….n components.

$P_j$ = Coefficient of components in term of price associated.

$b_j$ = Coefficient of components in term of best seller rating associated.

$d_j$ = Coefficient of components in term of download rating associated.

$r_j$ = Coefficient of components in term of review based rating associated.

Now this model can be implemented in popular optimization software package LINDO. The Implementation is shown in the next section.

## IV. An Illustrative Example Using Lindo

Consider a hypothetical problem where the developer is looking for the .Net components for managing the documents in an organization. The requirements can be stated as below:

i. The user is able to annotate the document by adding the comments, highlighting some text, images, tables etc.
ii. The user is able to merge the documents and can easily identify the difference between two versions of the same document.
iii. The user is able to retrieve information across the websites.

The developer will look for components satisfying these requirements. S(he) considers the free online repository www.componentsource.com for the components. The developers realizes that a vast set of components are available corresponding to the requirements on this website, s(he) is actually puzzled that which components to be selected based on the information that is provided on the site. By following the classification provided based on the requirements, the developer reaches at a point where a set of twenty components are available based on the stated requirements. Each component is provided with the following information:

1. Textual description of the functionality provided by the component.
2. The pricing of the components, like the individual cost or the cost of more than one licensed version of the component.
3. The components are rated based on the past purchases under the bestseller rating.
4. Each time a developer buys a component s(he) writes a review on it, based on the reviews also the components are rated.
5. The number of times the trial version are downloaded, this parameter is also taken into consideration when the components are ranked or rated.
6. The asset value which denotes the time and experience required if the component is developed inhouse.

The developer wants to select the component set which is optimal using all these criteria so based on his intuition/judgment/experience considers the component to be selected, in the following priority levels, by fixing the cost for the components and the number of components to be retrieved:

1. The selected component should be in the top five bestseller component.
2. The chosen component should be in the top ten components in the download based rating.
3. In the review based rating the chosen component should be among the top ten components.

This software component selection problem is a good candidate for solving with Goal Programming as goals are set for the objective values and he tries to achieve all these goals.

In this section we implement the model for a set of twenty document management components for .Net. taken from www.componentsource.com. Each component is denoted as $c_1, c_2\, c_3$ etc. Each component is characterized by its pricing, bestseller rating, download rating & review based rating. The table 1 shows the component details. The main constituents of the model are the decision variables, hard constraints, goal constraints & the objective function. These are explained as follows:

### A. Decision Variable

The decision variable $C_i$ for this model are twenty components, i = 1,2 3,….20.As the component is either accepted or rejected, zero-one goal programming is used.

### B. Hard Constraints

*According* to the model the hard constraints are on the budget and number of components, as depicted below:

$$157500c_1 + 157500c_2 + 113400c_3 + 157500\, c_4 + 157500\, c_5 + 2400\, c_6 + 6300\, c_7 + 4700\, c_8 + 31400\, c_9 + 44000\, c_{10} + 94400\, c_{11} + 62900\, c_{12} + 201600\, c_{13} + 56600c_{14} + 50300c_{15} + 34600c_{16} + 18800c_{17} + 50300c_{18} + 12500c_{19} + 40900c_{20} < 200000 \quad (8)$$

$$\sum_{i=1}^{20} Ci \quad (9)$$

### C. Goal Constraints

There are three goal constraints, that are considered according to the priorities named as priority 1, priority 2 and priority 3 goals. Each priority corresponds to bestseller rating, download rating and review based rating.

Table 1 .Net components for document management

| Component | Pricing | Bestseller Rating | Review Rating | Download Rating |
|---|---|---|---|---|
| C1GroupDocs.Comparison | 157,500 | 1 | 2 | 6 |
| C2GroupDocs.Assembly | 157,500 | 2 | 3 | 13 |
| C3Vizit | 113400 | 5 | 5 | 2 |
| C4GroupDocs.Annotation | 157,500 | 3 | 4 | 5 |
| C5GroupDocs.Viewer | 157,500 | 4 | 1 | 9 |
| C6Covri Parent Selector Column Web Part | 2400 | 6 | 6 | 17 |
| C7Covri Cascaded Lookup Web Part | 6300 | 8 | 8 | 8 |
| C8Covri CrossSite Lookup Web Part | 4700 | 7 | 7 | 18 |
| C9Virto Active Directory User Service for SharePoint | 31400 | 11 | 11 | 10 |
| C10,Virto Create & Clone AD User Web Part | 44000 | 10 | 10 | 14 |
| C11, Virto Workflow Activities Kit | 94400 | 9 | 9 | 15 |
| C12, jungle doc | 62900 | 17 | 17 | 7 |
| C13,KWizCom SharePoint Wiki Plus | 201600 | 12 | 12 | 16 |
| C14 SharePoint AD Information Sync | 56600 | 13 | 13 | 11 |
| C15,SharePoint Batch Check In | 50300 | 18 | 18 | 4 |
| C16,SharePoint Document Auto Title | 34600 | 19 | 19 | 20 |
| C17 SharePoint Document Number Generator | 18800 | 15 | 15 | 3 |
| C18,SharePoint Document Viewer | 50300 | 16 | 16 | 1 |
| C19 SharePoint Item Audit Log | 12500 | 14 | 14 | 19 |
| C20 SharePoint RichText Boost | 40900 | 20 | 20 | 12 |

- *Priority 1*

The software component to be selected should be among the top 5 bestseller components. Here the developer doesn't want overachievement with respect to the target.

$$1c_1 + 2c_2 + 5c_3 + 3c_4 + 4c_5 + 6c_6 + 8c_7 + 7c_8 + 11c_9 + 10c_{10} + 9c_{11} + 17c_{12} + 12c_{13} + 13c_{14} + 18c_{15} + 19c_{16} + 15c_{17} + 16c_{18} + 14c_{19} + 20c_{20} + y_1 - y_2 \leq 15 \quad (10)$$

- *Priority 2*

From the repository of 20 components the components to be selected should be among the top 10 of the number of download based rating.

$$6c_1 + 13c_2 + 2c_3 + 5c_4 + 9c_5 + 17c_6 + 8c_7 + 18c_8 + 10c_9 + 14c_{10} + 15c_{11} + 7c_{12} + 16c_{13} + 11c_{14} + 4c_{15} + 20c_{16} + 3c_{17} + 1c_{18} + 19c_{19} + 12c_{20} + y_3 - y_4 \leq 55 \quad (11)$$

- *Priority 3*

According to the reviews made by other users who already use this component, the ratings are done & on this basis the developer wants to have rating of top 10.

$$2c_1 + 3c_2 + 5c_3 + 4c_4 + 1c_5 + 6c_6 + 8c_7 + 7c_8 + 11c_9 + 10c_{10} + 9c_{11} + 17c_{12} + 12c_{13} + 13c_{14} + 18c_{15} + 19c_{16} + 15c_{17} + 16c_{18} + 14c_{19} + 20c_{20} + y_5 - y_6 \leq 55 \quad (12)$$

### D. *Objective Function*

The Goal Programming is to minimize the value of the objective function subject to goal constraint and satisfying the pre-emptive priority goals.

$$\text{Minimize } (y_2 + y_4 + y_6) \quad (13)$$

The above mentioned constraints and objective functions when executed on LINDO gives results as discussed in the subsequent section.

## V. Results And Discussion

In the software component selection problem either the component is selected or rejected so we are considering Zero-One Goal programming [17].

At the first run, $y_2$ is minimized and if the objective function value is zero that means the first priority is fully satisfied. As shown in the figure 1, all the target values are achieved. The reduced cost column shows that amount by which the objective coefficient of the variable would have to improve before it would become profitable to bring that variable into the solution at a nonzero value. There is a slack in the three goal constraints, which is tolerable according to our priorities. The initial set of components retrieved includes c1, c6 and c8.According to priority 1 goal constraint the sum of prices of three components is Rs 1,64,600 with a slack of Rs 35400 as shown in fig. 1.Similarly, for the download based rating the rating for the selected components is 6,17 and 18 respectively and the sum has a slack of 14 as shown in the fig.1.

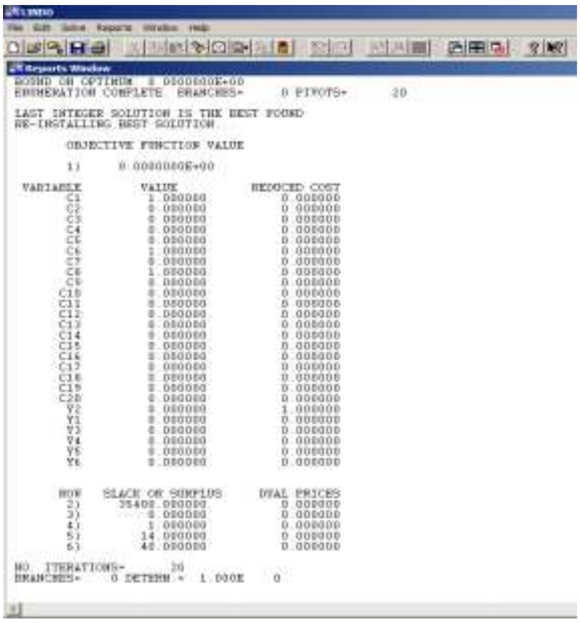


Fig. 1. The report window for minimizing $y_2$

In the second run, for minimizing $y_4$, the value of $y_2$ is added as to the set of constraints. On solving the model the results obtained are shown in fig. 2. That shows similar results to fig. 1 but the constraint $y_4$ has been minimized to zero.

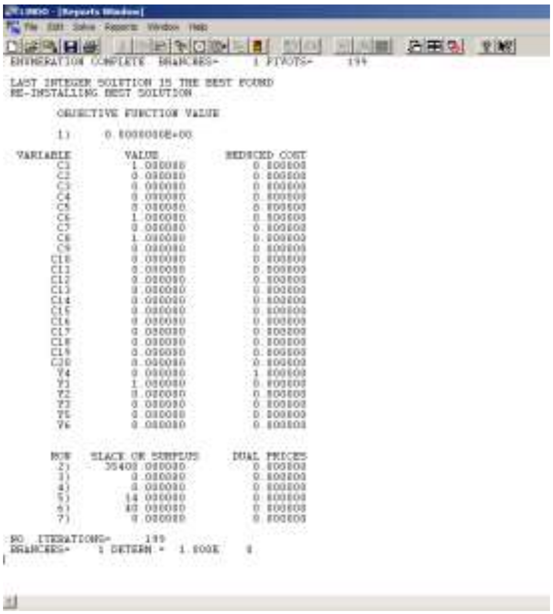


Fig. 2. The report window for minimizing y4

Subsequently, in the third and the final run the value of $y_2$ and $y_4$ are added to the set of constraints while minimizing $y_6$. It is depicted in fig. 3 that there is a slack in achievement of goal constraints but the hard constraints are fully met.

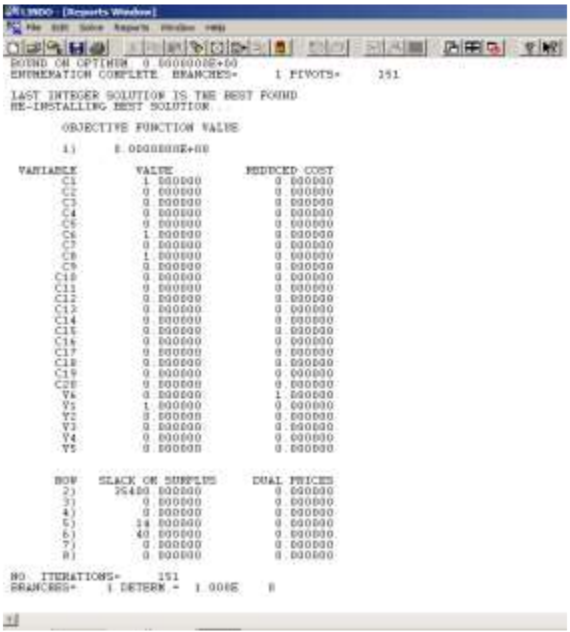


Fig. 3. The report window for minimizing y6

There is a zero slack for the bestseller based rating and 40 for the review based rating, The final component set is c1 (GroupDocs.Comparison), c6 (Covri Parent Selector Column Web Part) and c8 (Covri CrossSite Lookup Web Part).The retrieved components are according to the functionality desired like the component can easily make out the differences in the revised version of the document and it will be able to hierarchically the information across various websites.

## VI. CONCLUSIONS

The Component Based Software Development is the latest paradigm to achieve the reusability that results in low development cost and accelerated delivery. The software component selection is done based on the functionality. The component repository also provides some meta data to make the decision of selecting the best fit component. The proposed model attempts to find the best candidate components based on the user set priorities from an online repository. After mathematically formulating the problem, it provides interesting results based on the given goals and user preferences. All the rigid conditions are specified explicitly and the user defined priorities are also taken into consideration based on this the optimal component sets are generated. This approach gives satisfactory results when the component repository is small and the number of components to be retrieved are also less. But for larger repository this may not give optimal results in that case fuzzy clustering will be used which will be considered in the future work.

## REFERENCES

[1] I. Sommerville, *Software Engineering*. Pearson Education, 2009

[2] Brown, A. W., *Large-Scale, Component-Based Development*, Prentice Hall PTR,2000

[3] Szyperski C., (1998). Component Software, Beyond Object-Oriented Programming, ACM Press, Addison-Wesley, NJ.

[4] S.Kaliraj, N. Premkumar and A. Bharathi, "The Novel Life Cycle Model for Component Based Software System Based on Architecture Quality Using KCW Framework", *International Journal of Information Technology and Computer Science*,Vol. 9,pp. 74-79,Sept. 2014.

[5] N. Gehlot, and J. Kaur, "Dynamic inheritance coupling metric-design and analysis for assessing reusability", *Int. J. Software Engineering, Technology and Applications*,Vol. 1, No. 1, pp.118–133,2015.

[6] Yoon K. Paul, Hwang Ching-Lai, Multiple Attribute Decision Making: An Introduction ,Sage Publisher,1995.ISBN:0-8039-5486-7.

[7] Mollaghasemi, M., Pet-Edwards, J., 1997. Technical briefing: making multiple objective decisions. IEEE Computer Society Press, Los Alamitos, CA.

[8] Vescan, A., *Pareto dominance - based approach for the Component Selection Problem,* Proceedings of the 2nd UKSim European Symposium on Computer Modelling and Simulation, 8 - 10 September, Liverpool, England, ISBN: 978-0-7695-3325-4, pp. 58-63, 2008.

[9] Vescan A., Grosan C., Shengxiang Yang "A hybrid evolutionary multiobjective approach for the dynamic component selection problem", in Proceedings of 11th International Conference on Hybrid Intelligent Systems (HIS), pp 714-721, 2011 .

[10] Tang, J. F., Mu, L. F., Kwong, C. K., & Luo, X. G. (2011). An optimization model for software component selection

under multiple applications development, *212*, 301-311. doi:10.1016/j.ejor.2011.01.045

[11] Jha PC,Arora R, Kumar UD,"A Fuzzy Approach for Components selection amongst Different versions of Alternatives for a Fault Tolerant Modular Software System under Recovery Block Scheme Incorporating Build –or-Buy Strategy",2011,American Journal of Operation Research, pg 249-258.

[12] Yi Liu, Fuzan Chen, Minqiang Li, Jisong Kou, "A Multi-objective Optimization Model for Information System Design" in proceedings of International Conference, Tianjin, China,pp 486-495,2012.

[13] Jha P C and Bali V., "Goal Programming Approach for Selection of COTS Components in Designing a Fault Tolerant Modular Software System under Consensus Recovery Block Scheme",International Journal of Computer & Communication Technology, Vol.3,No. 1,pp.1-8.

[14] Ali I.,Singh Raghav Y. S. and Bari A., "Integer Goal Programming approach for finding a compromise allocation of repairable components", 2011, International Journal of Engineering Science & Technology, Vol. 3, No. 6,pp. 184-195.

[15] U. C Orumie and D Ebong, "A Glorious Literature on Linear Goal Programming Algorithms", *American Journal of Operation Research* , Vol. 4,pp. 59-71, 2014.

[16] S.Acharya, S. Nanday and B.B. Mishraz, " Solving Multi-Choice Linear Goal Programming Problem with Preemptive Priorities", http://www.academia.edu/2700366/Solving_Multi-Choice_Linear_Goal_Programming_Problem_with_Preemptive_Priorities.

[17] L. Schrage "Optimization modelling with LINDO" Cengage Learning, 1997.

## Authors' Profiles

**Jagdeep Kaur** working as Assistant Professor in Computer Science and Engineering Department, ITM University, Gurgaon, Haryana, India; and pursuing Ph.D. in Software Engineering from School of Information and Communication Technology, Gautam Budhha University, Greater Noida, UP, India. She holds a Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering from BCET, Gurdaspur, Punjab, India (2003). She obtained her Master of Technology degree in Computer Science from Department of Computer Science and Engineering, Punjabi University, Patiala India (2005). Her research interests include Component Based Software Engineering, Software Reuse, Software Testing and Software Process Metrics.

**Dr. Pradeep Tomar** is working as Faculty Member in the Department of Computer Science and Engineering, School of Information and Communication Technology, Gautam Buddha University, Greater Noida, Uttar Pradesh, INDIA since its inception in the year 2009. Dr. Tomar earned his Doctorate in Computer Science in the year 2011 under the supervision of a renowned academician and researcher - Professor N. S. Gill from Department of Computer Science & Applications, Maharishi Dayanand University, Rohtak, Haryana, INDIA. Before joining Gautam Buddha University he worked as Software Engineer in a multi-national company, Noida and lecturer in M. D. University, Rohtak, Haryana and Kurukshetra University, Kurukshetra, Haryana. Dr. Tomar has good teaching, research and software development experience as well as vast administrative experience at university level on various posts like research coordinator, examination and admission coordinator, time table coordinators, proctor and hostel warden. Presently he is also working as a Treasure in CSI, Noida Chapter. Dr. Tomar is also a member of IEEE, IEEE Computer Society, Computer Society of India (CSI), Indian Society for Technical Education (ISTE), Indian Science Congress Association (ISCA), International Association of Computer Science and Information Technology (IACSIT) and International Association of Engineering (IAENG). He served as a reviewer of journals and conferences and worked as advisory board members in national and international conferences. Dr. Tomar has qualified the National Eligibility Test (NET) for Lecturership in Computer Applications in 2003, Microsoft Certified Professional (MCP) in 2008, SUN Certified JAVA Programmer (SCJP) for the JAVA platform, standard edition 5.0 in 2008 and qualified the IBM Certified Database Associate - DB2 9 Fundamentals in 2010. Two books "Teaching of Mathematics" and "Communication and Information Technology" at national levels have been authored by Dr. Tomar. Dr. Tomar has been awarded with Bharat Jyoti Award by India International Friendship Society in the field of Technology in 2012. He has been awarded for the Best Computer Faculty award by Govt. of Pondicherry and ASDF society. He has been also awarded with best paper presentation award in 2013 2nd International Conference on Software and Computer Applications- ICSCA 2013 at Paris, France. His biography is published in Who's Who Reference Asia, Volume II. Several technical sessions in national and international conferences had been chaired by Dr. Tomar and he delivered expert talks in FDP, workshops, national and international conferences. Three conferences have been organized by Dr. Tomar: one national conference with COMMUNE group and two international conferences, in which one international ICIAICT 2012 was organized with CSI, Noida Chapter and second international conference 2012 EPPICTM was organized in collaboration with MTMI, USA, University of Maryland Eastern Shore, USA and Frostburg State University, USA at School of Information and Communication Technology, Gautam Buddha University, Greater Noida, Uttar Pradesh, INDIA. Apart from teaching, researches in the areas of Software Engineering are being guided by Dr. Tomar. He has also contributed more than 74 papers/articles in national/international journals and conferences. His major current research interest is in Component-Based Software Engineering.

**How to cite this paper:** Jagdeep Kaur, Pradeep Tomar,"Multi Objective Optimization Model using Preemptive Goal Programming for Software Component Selection", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.9, pp.31-37, 2015. DOI: 10.5815/ijitcs.2015.09.05