Soft Computing Particle Swarm Optimization based Approach for Class Responsibility Assignment Problem

Dinesh Kumar Saini
FCIT, Sohar University, Oman
Research Fellow,
Faculty of Engineering and IT,
University of Queensland,
Brisbane, Australia

Yashvardhan Sharma
Computer Science Dept., I.P.C.Unit, BITS Pilani,
Rajasthan, India

ABSTRACT

Soft computing approach like swarm optimization can be used in the software development environment also for developing effective software. Software development and testing cost must be reduced and optimized so that companies can survive and sustain in the market. Object oriented analysis and design is used for developing complex real time software systems. In object oriented software methods, class plays very crucial role and all the systems responsibility and functional and non-functional requirements are implemented through class. Assigning responsibility of the class with optimization is an issue which should be dealt carefully. An attempt is made to study Class Responsibility Assignment in the context of object oriented analysis and design. It is a crucial issue encountered in the software design phase in the software development life cycle. Class Responsibility Approach (CRA) depends on human judgment and decision making skills to a great extent. In this paper we have presented an algorithm using Particle Swarm Optimization to provide decision-making support for class responsibility assignment, to re-assign methods and attributes to classes in a class diagram. This will help the designers and developers of the object oriented software.

General Terms

Software Systems, Software Engineering Techniques, Object Orientation, Soft Computing.

Keywords

Particle Swarm Optimization (PSO), Class Responsibility Assignment (CRA), Genetic Algorithm (GA).

1. INTRODUCTION

Software engineering is a systematic approach for developing high-quality software in a cost-effective manner [1,3]. The development of software consists of feasibility study, analysis, design, implementation, testing, installation and maintenance. This is called software development life cycle (SDLC), which deals with the software right from its inception to its retirement [2]. Real world objects are modeled through Object Oriented Analysis and Design which is the software engineering approach widely used these days [4]. It aims at design and analysis of objects in which the most crucial task is to assign responsibilities to the identified classes. Class Responsibility Assignment deals with assigning responsibilities to the classes identified from the problem statement and how they interact with each other. CRA is a difficult task and depends on the skill and experience of the designer, to a great extent. CRA is a crucial and challenging task these days. Hence various soft computing techniques like Genetic Algorithm [5] are used in order to provide better decision making support. Particle Swarm Optimization is a meta-heuristic technique widely used in the field of software engineering for optimizing several issues. It is inspired by swarm intelligence, like how flock of birds converges quickly and together, to the source of food .The following sections discusses the related works in this field, proposed strategy to apply PSO to optimize the solution, analysis of the proposed strategy, the results obtained on applying this on a case study. Then these results are compared with those obtained from existing methods. Finally future scope for this work is discussed.

2. BACKGROUND

Various soft computing techniques like multi-objective genetic algorithms [5] have already been used for solving the class responsibility assignment problem in object oriented analysis. Simulated annealing algorithm [8] has been used to automatically improve the structure of an existing inheritance hierarchy. Also using a GA Search Based Determination of Refactoring for Improving the Class Structure in OO Systems is done [9].

However, Particle Swarm Optimization is known to have some advantages over these techniques like less sensitivity to the nature of objective function, etc as discussed later[15,16,17]. So, PSO algorithm has been used to optimize the class responsibility. Moreover CRA problem and PSO are similar in many aspects. First of all, CRA must have a very large search space which is also the case with PSO. There is no known efficient and complete solution in CRA; PSO also provides a set of solutions instead of a unique final solution. Finally, both PSO and CRA require the formulation of a suitable fitness function. Most of the complex and big systems are developed using object oriented approach and classes design and their responsibilities must be fixed properly.

3. PARTICLE SWARM PTIMIZATION (PSO)

Particle swarm optimization (PSO) which is an optimization technique based on social behavior of bird flocking or fish schooling [6]. PSO has many similarities with evolutionary and intelligent computation techniques like Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation [7]. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Stability of the object oriented class level is studied by Bouktif and quality predictive modeling is proposed [18]. The predictive model takes the form of a function that receives as input a set of structural metrics and an estimation of stress, and produces as output a binary estimation of the stability. Here, stress represents the estimated percentage of added methods in a class between two consecutive versions [23].

3.1 Definitions and Concept of PSO

Swarm - An initial population of particles is to be considered for applying PSO. This is called swarm. The selection of particles and population depends on the problem we are trying to optimize [11, 12, 13, 14].

Particles- An individual entity in the population is called as a particle.

Fitness Function – A fitness function is to be chosen according to the objective. Each particle will evaluate the fitness individually in each iteration.

Local Best – The best solution attained by a particle in its life time is called as a local best.

Global Best- The best of all local bests in the whole population is considered as the global best.

Iterations- PSO is applied in various iterations, in each the position and velocity of the particle is updated. Local best and global best are also updated in each iteration.

Optimal Solution – Finally the entire swarm will converge to a solution which will be optimal.

3.2. The PSO algorithm

```
Algorithm
Input: Randomly initialize position and velocity of the particles
Output: Position of the approximate global optima
Begin
While terminating condition is not reached do
Begin
For i = 1 to number of particles
Evaluate the fitness of each particle
Update local best and global best
Adapt velocity of the particle using equation
    $vij(t+1) = vij(t) + c1R1 (pij(t)-xij(t)) + c2R2 (pgj(t)-xij(t))$
Update the position of the particle using equation
    $xij(t+1) = xij(t) + vij(t+1),$
increase i;
end while
```

vij and xij are vectors representing the velocity and position of the particle respectively. c1, c2 are constants representing cognitive and social parameters, respectively. The combination of these two parameters determines the property of convergence of the algorithm. The portion of the adjustment to the velocity influenced by the local best is considered to be the cognitive component, and the portion influenced by the global is the Social component.

R1, R2 are random numbers between 0 and 1.

pij(t) is the local best, which is the best value attained by an individual particle till the time t

pgj(t) is the global best, which is the best of all values obtained in the entire swarm till the time t.

In each iteration, the position and the velocity of the particle as well as the local best and global best are updated according to the above equations.

After a few iterations all the particles will converge to an optimal solution.

4. THE CLASS RESPONSIBILITY ASSIGNMENT PROBLEM

The class responsibility assignment (CRA) problem deals with the ambiguities in assigning responsibilities to various classes in a class diagram and their interactions with each other. CRA problem is encountered in the early stages of object oriented software design [4]. CRA is about deciding where the responsibilities (class operations as well as the attributes they manipulate), belong and how the objects should interact (by using those operations).

The below example the Monopoly game model [3, 4] shows an illustration of the CRA problem ,Suppose two alternative CRAs for Monopoly: operation roll() is assigned to Player (Figure 1(a)) or Die (Figure 1(b)); note that roll() uses attributes MAX and face Value. Analyzing these alternative CRAs in the light of the general responsibility assignment software patterns (GRASP)[4], a collection of patterns to support object-oriented design.

According to the Information Expert pattern, responsibility to a class that has the information (attribute) needed to fulfill it: operation roll() should therefore be in class Die (where MAX and face Value are located). According to the Low Coupling pattern, one should assign responsibilities so that coupling remains low: Figure (b) shows higher class coupling than Figure1 (a) since roll() in Player has to ask Die for attribute

Values (MAX and face Value).

According to the High Cohesion pattern, one should assign responsibilities so that cohesion remains high:

Figure1(b) shows lower cohesion than Figure1(a) since roll() is functionally related to MAX and face Value and therefore the three should be in the same class.

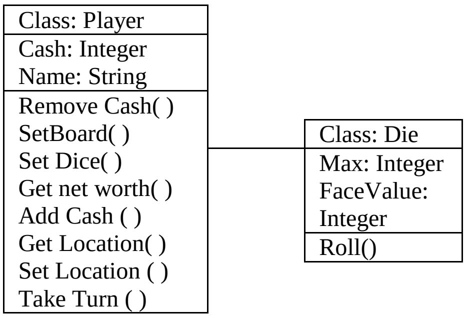

Figure 1(a).

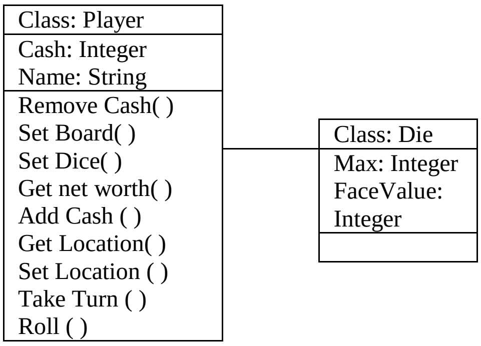

Figure 1(b)

5. PROPOSED STRATEGY

This work has been carried out to provide a decision making support for class responsibility assignment problem by considering various parameters like coupling, cohesion etc. To achieve this aim, Particle Swarm Optimization is applied on an objective function which considers various parameters which are explained in detail.

MAC(c) - method-attribute coupling between class c and other classes – Total number of interactions between methods of one class (c) and the attributes of other classes in the same class diagram

MMC(c) - method-method coupling between class c and other classes– Total number of interactions between methods of a class (c) and methods of other classes in the same class diagram

MGC(c) method-generalization coupling – This is applicable only in those class diagrams where generalization is applied.

5.1 Cohesion Measures

RCI(*c*) ratio of cohesive interactions of class

c : $RCI(c) = |CI(c)| / |CImax(c)|$.

CImax(c) is the maximum possible cohesive interactions in a class c. CI(c) is the number of existing cohesive interactions in a class c.

TCC(*c*) tight class cohesion - is the pairs of methods, m1 and m2, of a class c∈C with common usage.

5.2 General Measures

DMA (m,a) direct method–attribute dependency between method m and attribute a.

DMM (m1,m) direct method–method dependency between m1 and m2 .

Considering the above parameters the below equation [3] was derived.

$$F(C)= w_1RCI(C)+W_2TCC(C)- W_3(MAC(C)/DMA(C)- W_4(MMC(C)/ DMM(C))-W_5((MGC(C)/ DMA(C)+DMM(C))$$

Formula 1

According to the Figure 2, the input to the system will be class diagram and sequence diagrams. It is assumed that the user is an experienced developer. The user can input the range of values of various parameters based on his experience. Since the interactions between various classes can be illustrated better using a sequence diagram, the system accepts that also along with the class diagram. But still, evaluating sequence diagrams and estimating values for various parameters is beyond the scope of this paper.

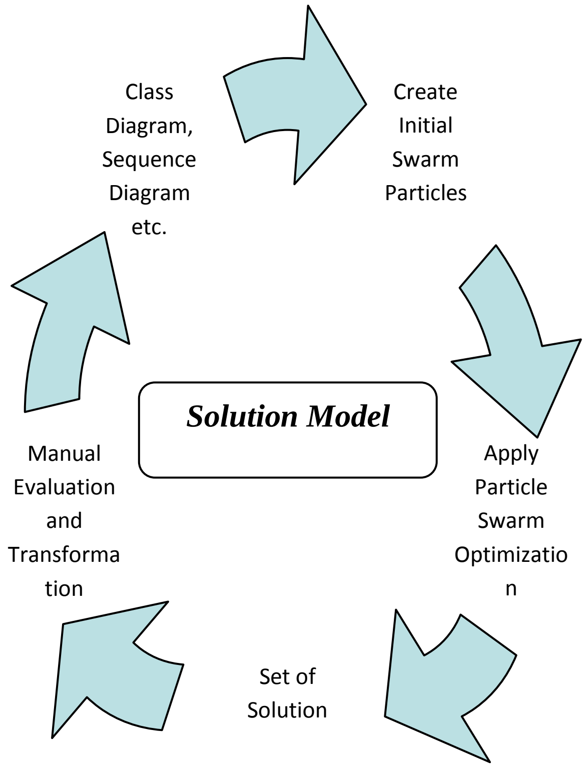
Class Diagram, Sequence Diagram etc.
Create Initial Swarm Particles
Solution Model
Apply Particle Swarm Optimizatio n
Set of Solution
Manual Evaluation and Transforma tion

Figure 2 Solution Model

The solution model flowchart explains the strategy followed in the optimization process. The steps included are initialization of swarm, application of PSO algorithm and analysis of the obtained output [18,19, 24,25]

The steps are explained below in detail:

1. A search space of n particles is considered. Here particles are different versions of a class diagram with randomly assigned attributes, operations and associations.

2. Each particle is represented as a function of various parameters of the objective function. The objective is to maximize the fitness function.

3. These particles will move in search space to achieve global best according to PSO equations [7]:

Velocity of the particle is updated by the equation:

$v_{ij}(t+1) = v_{ij}(t) + c_1R_1\ (p_{ij}(t)-x_{ij}(t)) + c_2R_2\ (pg_j(t)-x_{ij}(t))$

Position of the particle is updated by the equation:

$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$.

According to these equations, the solution will move in the search space trying to reach the optimal value in each iteration. These equations are applied iteratively until particles converge to optimal solution. In every iteration, each particle has its local best (best position achieved by that particle in its lifetime). Best position from all the local best is the global best and movement of particles is dependent on both local and global best.

4. After a few iterations all the particles will converge to an optimal solution. Based on the values of all the parameters in

the final set of solutions, optimized class diagrams are redrawn.

5. As an output, an optimized class diagram can be chosen from the set of solutions obtained.

6. ANALYSIS OF PROPOSED APPROACH

Applying PSO helps in optimizing the class responsibility assignment. Other heuristic and soft computing approaches like Artificial Intelligence, Neural Networks, etc can also be used for this purpose but PSO was opted because it has certain advantages over these techniques, which can be stated as under. [15,16].

1. PSO can be easily understood and implemented in comparison with other techniques for optimization.
2. In PSO, sensitivity to the nature of objective function is less.
3. It has limited number of parameters (inertia weight factor and two acceleration coefficients) in comparison with other optimization methods.
4. Impact of parameters to the solutions is less when compared to other heuristic algorithms.
5. It is less dependent on initial points in the search space.
6. It gives high quality and stable results in lesser time.

   So, PSO ends up giving accurate results in less time and effort.

Though there are lot of advantages in applying PSO, there are few constraints also which are listed below:

1. The results are not very stable, that is, the results obtained may not be always closer to the actual effort.

2. Results are dependent, to some extent, on values of the variants.

PSO helps in taking the architectural decision with the help of search based techniques. It will help the object oriented systems to decide the architectural style and design pattern.

7. CASE STUDY

We are considering a simple case study of a dice game which has two classes
Player
Die
Random variants of the class diagram are taken as particles.
Three examples of random assignment of particles and the respective evaluation of the objective function are shown below in Figure 3, 4 , 5.
For Particle 1 (Figure 3),

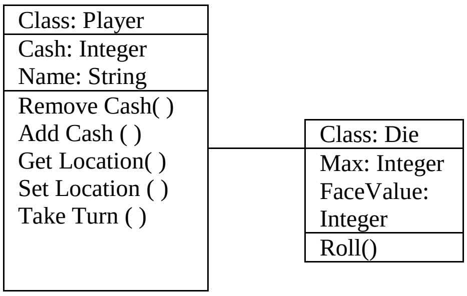

Figure 3

Calculating various parameters of the objective function from the above diagram, the following values were obtained.
MAC(Die)=0 since none of the methods in Die are using any attributes of Player.
MMC(Die)=1 since setDice( ) method in Die is using roll() method in Player. Similarly we obtained the values for the following parameters.
MAC(Player)=MMC(Player)=2
RCI(Die)=0 RCI(Player)=.3
DMA=3+2=5
DMM=0
Using the *Formula 1* the fitness function was evaluated to be
F(C)=TENDS TO NEGATIVE INFINITY

For particle 2 (Figure 4),

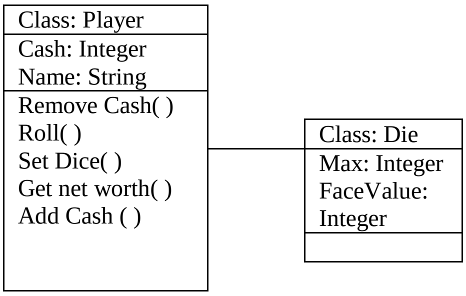

Figure4

MAC(Die)=MMC(Die)=0
MAC(Player)=2
MMC(Player)=0
RCI(Die)=0 RCI(Player)=.3
DMA=3
DMM=3
TCC=3
F(C)=.54

For Particle 3 (Figure 5),

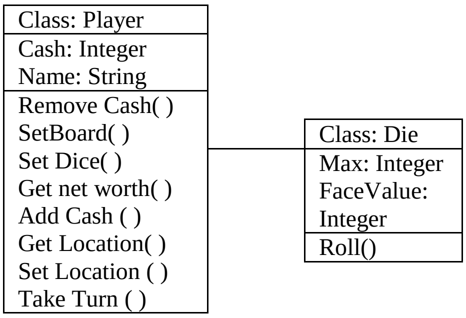

Figure 5.

MAC(Die)=MMC(Die)=0
MAC(Player)=1
MMC(Player)=1
RCI(Die)=1 RCI(Player)=.375
DMA=5
DMM=2
TCC=3
F(C)=.735

After carrying out the evaluation of fitness function for 10 iterations, it was found that the solution converged to 0.75. Hence it was concluded that Figure 5 is the optimal solution. So particle 3 is initial global best and all the values of particular parameters are local best. Using these initial values iterations are applied on these particles till they converge to optimal solution.

8. CONCLUSION AND FUTURE WORK

By applying PSO on CRA it was concluded that the solution need not always be the best but it will be optimal.Possibility of further optimization can be explored by applying other variants of PSO. Possibilities of deriving the best solution are under study. In the object oriented software system every class should have set of responsibility and it should be allocated optimally. We can find the optimal fitness function for each software component.

9. ACKNOWLEDGEMENT

The Author is very thankful to the reviewers for giving important feed for the manuscript. Author would also like to thank Sohar University and University of Queensland for their support and encouragement.

10. REFERENCES

[1] Dinesh Kumar Saini and Moinuddin Ahmad "Modeling of Object Oriented Software Testing Cost" The 2011 International Conference on Software Engineering Research and Practice (SERP'11), World Congress in computer Science and Engineering, July 18-21, 2011, Las Vegas, USA. Pp. 333-339.

[2] Dinesh Kumar Saini and Moinuddin Ahmad "Enhanced Software Quality Economics for Defect Detection Techniques Using Failure Prediction" The 2011 International Conference on Software Engineering Research and Practice (SERP'11) World Congress in computer Science and Engineering July 18-21, 2011, Las Vegas, USA, PP. 346-351

[3] Ian Sommerville," Software Engineering", Eighth Edition, Pearson Edition, 2009.

[4] Lingaraj A. Hadimani, Dinesh Kumar Saini, Vaishali P Khoche and Sanad Al Maskari, "Comparison of Software and Hardware Design Tools (CASE vs. Simulators)" The 2011 International Conference of Manufacturing Engineering and Engineering Management, (ICMEEM-2011), World Congress in Engineering, July 6-9th London UK

[5] Jalote , P. "An Integrated Approach to Software Engineering", Third Edition, New York: Springer Science+Business Media, Inc., 2005, p. 16.

[6] Bowman M., Briand L. C. and Labiche Y., "Multi-Objective Genetic Algorithms to Support Class Responsibility Assignment," Carleton University, Technical Report SCE-07-02 version 3, http://squall.sce.carleton.ca, 2008.

[7] Clarke, J, et al., "The Application of Metaheuristic Search Techniques to Problems in Software Engineering",SEMINAL-TR-01-2000: August 30, 2000.

[8] Dinesh Kumar Saini and Hemraj Saini "Static Code Analysis", Proceeding of National Seminar on Mathematics and Computer Science sponsored by UGC.

NSCOMCS-2005

[9] Larman C., Applying UML and Patterns, Prentice-Hall, 3rd Edition, 2004.

[10] Bowman M.,et al, "Solving the Class Responsibility Assignment Problem in Object-oriented Analysis with Multi-Objective Genetic Algorithms," IEEE Transactions on Software Engineering, 23 Jul. 2010.

[11] Dinesh Kumar Saini and Hemraj Saini "Statistical Modeling of Extensibility in software" 3rd International Conference on Quality, Reliability and INFOCOM Technology (Trends and Future Directions), 2-4 December, 2006, Indian National Sciences and Academics, New Delhi (India). ISBN 81–7446–434–4 Conference proceeding.

[12] The PSO website. (2006) [Online]. Available: http://www.swarmintelligence.org/.

[13] Parsopoulos, K.E, & Vrahatis, M.N., Particle Swarm Optimization and Intelligence: Advances and Applications, New York: Information Science Reference, 2010.

[14] O'Keeffe M. and O Cinneide M., "Towards automated design improvement through combinatorial optimization," Proc. Workshop on Directions in Software Engineering Environments, 2004.

[15] Seng I., Stammel J. and Burkhard D., "Search-based determination of refactorings for improving the class structure of object-oriented systems," Proc. Conference on Genetic and Evolutionary Computation, 2006.

[16] Dinesh Kumar Saini and Nirmal Gupta "Class Level Test Case Generation in Object Oriented Software Testing, International Journal of Information Technology and Web Engineering, (IJITWE) Vol. 3, Issue 2, pp. 19-26 pages, march 2008. USA

[17] Dinesh Kumar Saini and Hemraj Saini "VAIN: A Stochastic Model for Dynamics of Malicious Objects", the ICFAI Journal of Systems Management, Vol.6, No1, pp. 14- 28, February 2008. INDIA

[18] S. Bouktif, B. Kégl, H. Sahraoui, Combining software quality predictive models: an evolutionary approach, in:Proceedings of the International Conference on Software Maintenance, ICSM'02, 2002, pp. 385–392.

[19] Hemraj Saini and Dinesh Kumar Saini "Malicious Object dynamics in the presence of Anti Malicious Software" European Journal of Scientific Research ISSN 1450-216X Vol.18 No.3 (2007), pp.491-499 © Euro Journals Publishing, Inc. 2007 http://www.eurojournals.com/ejsr.htm EUROPE

[20] Dinesh Kumar Saini and Hemraj Saini "Static Code Analysis", NSCOMCS-2005 Proceeding of National Seminar on Mathematics and Computer Science sponsored by UGC.

[21] Dinesh Kumar Saini and Hemraj Saini "Identification and characterization of software testing process for object oriented systems", National Conference on Mathematical Analysis and its Applications in Real - World Problems, Berhampur University, September

[22] S. Bouktif, D. Azar, H. Sahraoui, B. Kégl, D. Precup, Improving rule set based software quality prediction: a genetic algorithm-based approach, Journal of Object Technology 3(4) (2004) 227–241.

[23] Dinesh Kumar Saini and Hemraj Saini "Software Metrics and Mathematical Models in the Software Development Environment for Improving its Quality", National Conference on Mathematical Modeling, BITS Pilani, Oct.2005

[24] Dinesh Kumar Saini and Hemraj Saini "Statistical Modeling of Extensibility in software" 3rd International Conference on Quality, Reliability and INFOCOM Technology (Trends and Future Directions), 2-4 December, 2006, Indian National Sciences and Academics, New Delhi (India). ISBN 81–7446–434–4 Conference proceeding.

[25] Dinesh Kumar Saini, Lakshmi Sunil Prakash and Wail M Omar "Review of Technological Challenges in Web - Based Learning Content Management Systems (LCMS) with special emphasis on extraction of Learning Contents" International Symposium, College of Applied Science, Ministry of Higher Education, April 13-16, 2010, Oman, P.No. 4-14.