ISSN No:-2456-2165

Adopting Particle Swarm Optimization Approach for Software Process Through the Aging Leader Algorithm

Neyole Misiko Jacob
Jomo Kenyatta University of Agriculture and Technology

Abstract:- Software engineering industry has been characterized by rapid innovation and intense competition. SE firms to survive must develop high quality software products on time and at low cost. a critical issue is whether high levels of quality software products can be achieved without adversely impacting cycle time and software engineers' development effort. Conventional beliefs hold it that SE processes to improve software quality can be implemented only at the expense of longer cycle times with more considerable development effort. However, an alternative view is that software quality improvement, fast cycle time, and software developers' effort reduction can simultaneously be attained by adopting particle swarm optimization approach for software processes through the understanding aging leader algorithm. The study empirically investigates the impact of aging leader algorithm to SE process maturity, quality, cycle time, effort from the developers and SE product outcome by major IT firm and how the effects can be optimized through the particle swarm optimization approach.

Keywords:- *Software Engineering (SE) processes, Particle Swarm Optimization, Aging Leader Algorithm.*

I. INTRODUCTION

The concept of Particle swarm optimization (PSO) can be described as a computational approach that optimizes problems by iteratively trying to improve a candidate (particle) solution about a given measure of quality. Wei-Neng Chen et al. [1] discussed that the algorithm tries to solve a problem by having a population of candidate solutions, where the candidates are dubbed particles. The particles are usually moved around in the search-space according to simple mathematical formulae over the particle's position and velocity.

In the PSO algorithm, each particle's movement is influenced by its local best-known position as it is guided toward the best-known positions in the search-space. Each of the new best-achieved positions is subsequently updated by other particles that find better positions. These particles are expected to move the swarm toward the best solutions. For significant problems in software engineering, many locally optimal solutions have to be obtained as a reasonably assured optimal solution, here an efficient reduction scheme needs to be incorporated into the program to reduce the total computation time by a substantial amount [2], and this is primarily applicable through the principle of PSO algorithm.

According to Jose Vina [3], every organism grows older with time, the process of being older (aging) makes organisms weak and unable to lead the population. These process then calls for the other young organisms to make themselves available for becoming the new leader as they rise as the challengers for the older leaders in a population. The concept of Aging is essential to maintain the diversity [4] where there may be some challenges available for taking up the leadership, but out of them, only one can lead the specific population. [1]. This Aging mechanism can be applied to various software engineering problems, and optimal solutions can be found [4].

II. SOFTWARE ENGINEERING (SE) PROCESSES

Over the past decades, IT firms have been investing heavily in software as information technology (IT) infiltrates and play a critical role in all aspects of the value chain [5]. As a result, the IT industry has experienced more than 24% higher than the average on the job outlook growth worldwide as at May 2016. On the other hand as the number of software developers and firms increases, competition intensifies with the median annual wage for a software developer, applications being $100,080 in May 2016 and the median annual wage for software developers, systems software being $106,860 in May 2016 [6].

Broadly, there are two primary processes which all other processes take precedence, these are the sequential models, that emphasize on movement from one phase to another sequentially and the evolutionary process models, which doesn't depend on the conclusion of one phase before moving to the next as it involves iteration [7]. Each of this phases approaches are limited to factors that might inhibit the execution of a software product such as budget constraints and time. The need to employ PSO methodology over the aging algorithm thus seem plausible.

From the assessment of challenges experienced during the deployment of software products, many software production firms have embarked on software process improvement (SPI) initiatives. The critical to the success of these initiatives is the transfer of knowledge across individuals who occupy a range of roles within various organizational units involved in software production [8]. SE as a process encapsulates a set of sequential practices that are functionally coherent and standardized for easy reusability, implementation, and maintenance by SE-firms

III. PARTICLE SWARM OPTIMIZATION

Qinghai Bai [9], noted that the main idea of the principle of PSO is that it presents the advantages for resolving economic dispatch (ED) problem which is considered as one of the severe problems to be tackled. According to Burak Akat and Veysel Gazi [10], particle swarm optimization (PSO) algorithm is a direct search optimization strategy that is non-gradient in which a population of particles searches in parallel for a minimum of a function in a multi-dimensional space. Qinghai [9] further concluded that since PSO is based on the intelligence, it can be applied to both scientific research and engineering use the study further accented that the algorithm does not overlap and has no mutation calculation hence the search can be carried out by the speed of the particle.

The PSO algorithm outline is based on the algorithm described by Kennedy and Eberhart [11], using modifications suggested in Mezura-Montes and Coello Coello [12] and Pedersen [13]. It begins by creating first particles and assigning them initial velocities. It then evaluates the objective function at each particle location and determines the best function value and location. It then chooses new velocities, based on the current velocity, the particles' individual best locations, and the best locations of their neighbors. It then iteratively updates the particle locations that is the new location is the old one plus the velocity, modified to keep particles from bounds, neighbors, and velocities. Iterations hence proceed until the algorithm reaches a stopping criterion.

Ke-Lin and Swamy [14] argue that PSO can locate the region of particle optimization faster, but once in this region it progresses slowly due to the fixed velocity step size. During the development of several generations, only the most optimist particle can transmit information onto the other particles, and the speed of the researching is breakneck among other advantages. Basing on this, PSO in SE process and projects can be used to facilitate best approaches to management to avoid problems of succession plan where the best successors can takeup a process in parallel to those reaching termination or failure.

IV AGING LEADER ALGORITHM

Wei-Neng Chen et al. [1] stated that in a social animal colony, through aging the old leaders of the colony weaken and thus provides opportunities for the other individuals to challenge the leadership position. The inspiration here is that as age catches u p with the older leaders of a team, the more energetic younger leaders take up the challenge to fit the position otherwise succession plan. The best challenger here, therefore, stands a chance to take the mantle of leadership from the older leader. By cons idering aging as a page replacement algorithm [15], this algorithm can be employed to as a succession plan for SE project management.

The constraint aging expressed as a measure to SE process extinction regardless of the success or failure of the process, aging as a factor should be considered to overhaul the process. In SE process, therefore, aging can be used as a scheduling technique for the avoidance process starvation. Here, aging algorithm can be used to ensure that jobs with priority complete to their execution. In SE process evaluation many techniques and tools have been tried and failed to deliver substantial global improvements to software evolution process [16], aging as a perceived shortcomings present SE with the need to speed up processes and p ut a succession plan in place to facilitate execution of software projects.

The most critical and possibly crucial role of the aging algorithm in SE process should be to enhance adequate theory and practice for process improvement to save on time and cost over-runs due to delays in project deployment. Further, still, the approach can be embodied in several models for SE processes, where each model describes approaches to a variety of tasks or activities that take place during the SE-process [17]. Much as some models support rapid delivery and flexibility, user involvement and looping as their high points, and others focus more on cost -effectiveness and meeting user requirements, all should consider aging as a factor that affects SE process.

V. CONCLUSION

Particle Swarm Optimization approach for software process through the aging leader algorithm approach is characterized by assigning the leader of the swarm with a growing age and a lifespan to allow other individuals to challenge the leadership when the leader becomes aged during SE process execution. When an SE process leader shows strong leading power, his lives longer to attract the swarm toward better positions. However, during the process, if a leader fails to improve the swarm and gets old, new particles should emerge to challenge and claim the leadership to brings in diversity in SE process. High performance is therefore confirmed through succession.

REFERENCES

[1]. W.-N. Chen, J. Zhang and H. S. H. Chung, "A Novel Set Based Particle Swarm Optimization Method for Discrete Optimization Problems," IEEE Transactions on Evolutionary Computation, vol. 14, no. 2, p. 278–300, 2010.

[2]. S. Lin, "Computer solutions of the traveling salesman problem," IEEE- The Bell System Technical Journal, vol. 44, no. 10, pp. 2245 - 2269, 1995.

[3]. J. Viña, C. Borrás, and J. Miquel, "Theories of aging," IUBMB Life Journals, vol. 4, no. 5, pp. 249-254, 2007.

[4]. A. Kaur, "Aging Theories and the Aging Leader Algorithm with Challenger," International Journal of Computer Science and Information Technology Research, vol. 3, no. 1, pp. 130-134, 2015.

[5]. D. E. Harter, M. S. Krishnan, and A. S. Sandra, "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," in Institute for Operations Research and the Management Sciences (INFORMS), Maryland, USA, 2000.

[6]. United States Department of Labor, "Occupational Outlook Handbook," USA Bureau of labor statistics, 24th October 2017. [Online]. Available: https://www.bls.gov/ooh/computer-and-information technology/software -developers.htm. [Accessed 24th September 2017].

[7]. I. Crnkovic, M. Chaudron, and S. Larsson, "Component based Development Process and Component Lifecycle," In Software Engineering Advances, International Conference on IEEE, pp. 44-44, 2006.

[8]. S. A. Slaughter and L. J. Kirsch, "The Effectiveness of Knowledge Transfer Portfolios in Software Process Improvement: A Field Study," Information Systems Research (ISR), vol. 17, no. 3, pp. 301 - 320, 2006.
[9]. B. Qinghai, "Analysis of Particle Swarm Optimization Algorithm," Computer and Information Science- CCSE, vol. 3, no. 1, pp. 180-185, 2010.
[10]. S. B. Akat and V. Gazi, "Decentralized asynchronous particle swarm optimization," In Swarm Intelligence Symposium IEEE, pp. 1-8, 2008.
[11]. J. Kennedy and E. R, "Particle Swarm Optimization," in Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1995.
[12]. E. Mezura-Montes and A. C. C. C, "Constraint-handling in nature -inspired numerical optimization: Past, present and future," p. 173–194, 2011.
[13]. M. E. Pedersen, "Good Parameters for Particle Swarm Optimization," Hvass Laboratories, Luxembourg, 2010.K. Du and M. N. Swamy, "Particle Swarm Optimization," In Search and Optimization by Metaheuristics, pp. 153-173, 2016.
[14]. H. Wang, Z. Fang, L. Liu, P. Xu and Q. Zhao, "The Improved File Caching Mechanism Based on Aging Algorithm," Advances in Information Technology and Industry Applications, vol. 136, pp. 531-538, 2012.
[15]. E. Gray and W. Smith, "On the limitations of software process assessment and the recognition of a required re orientation for global process improvement," Software Quality Journal, vol. 7, no. 1, p. 21–34, 1998.
[16]. S. Misra, M. Omorodion, and L. Fernandez-Sanz, "Overview of Software Process Models, their Benefits, and Limitations," Researchgates, 15th February 2016. [Online]. Available: https://www.researchgate.net/publication/293107406. [Accessed 25th October 2017]