

Research Article

Utilizing Virtual Crowd for Global Software Development

Ying Zhang,¹ Farhad Ali ,² Kunhao Wang ,³ Shah Nazir ,² and Zeqi Leng³

¹School of Information Engineering, Changchun University of Finance and Economics, Changchun 130122, China

²Department of Computer Science, University of Swabi, Swabi, Pakistan

³College of Computer Science, Jilin Normal University, Siping 136000, China

Correspondence should be addressed to Kunhao Wang; wkhoyoyo@jlnu.edu.cn and Shah Nazir; snshahnzr@gmail.com

Received 2 June 2021; Accepted 6 July 2021; Published 16 July 2021

Academic Editor: Zhongguo Yang

Copyright © 2021 Ying Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software firms are interested in outsourcing and developing of software globally to the virtual crowd for minimizing the product cost and for increasing the software quality. Developments in information technology (IT) have changed the organizational working environment from centralized to disperse development working practices. As a result, companies have recognized the value of virtual world networks that offer benefits such as efficient time management, lower cost of growth, reduced travel costs, and access to larger competent team members to select the right skilled individual. With the wide spread of Web 3.0 applications and improvements in cloud computation technologies, multinational, multiskilled, and diverse crowds carry out the software developmental process. The aim of this research is to select the effective virtual crowd for the development of quality software. The proposed “characteristic-based virtual crowd selection (CBVCS)” method will select the crowd according to their unique characteristics such as their skills, experiences, expertise, and knowledge.

1. Introduction

Professional software development is usually a collaborative effort that is classified as a socio-technical practice. The socio-technicality nature has increased the popularity of outsourcing and offshoring software development in recent years [1]. Innovations in technologies have significantly reshaped the software development environment [2]. With the wide applications of Web 3.0 and improvements in cloud computation, the software of this era are built with the help of the virtual crowd present on various sites on Internet. The global developmental processes have been driven by global virtual teams (GVTs). A member works without cultural, geographical, and time limits in GVTs [3]. In economic progression, the need for fast-to-market, low-cost, and quick solutions to complex organizational problems is increasingly driven by virtual teams. Virtual teams enable organizations to combine the skills and abilities of employees and non-employees to minimize time and space barriers. Firms now invest heavily in the virtual team to improve their performance and productivity [4]. Various software development methodologies have been adopted for developing software

[5–7]. Globalization is considered as a major trend especially among software developing industries. National markets are constantly transformed into global markets by economic forces, generating new forms of competition and cooperation across national borders. This change has far-reaching consequences not just for marketing and distribution, but also for the production, design, development, testing, and distribution of products. Recently, attention has been focusing on trying to understand the factors that enable multinationals and virtual firms to work efficiently across geographical and cultural boundaries. Almost every corporation now depends on software, as the success of organizations highly depends on utilizing the software as a strategic tool. The organizations have started the outsourcing and development of software globally for minimizing the costs and for obtaining access to specialized resources [8].

The cost of Internet access has decreased significantly as a result of developments in information technology (IT), resulting in a change from centralized to dispersed development working practices. As a result, companies have recognized the value of virtual world networks that offer

benefits such as efficient time management, lower cost of growth, reduced travel costs, and access to larger competent team members to select the right skilled individuals [9]. Software development necessitates a great deal of cooperation and coordination. It brings all of the software engineers together in one place, encouraging them to discuss goals and logically complete the projects [10]. GSD is thought to allow the continuous development of software by taking advantage of temporal differences between software engineers in different parts of the world [11]. For technological firms, global software development (GSD) is increasingly becoming the trend [12]. The organization utilizes the collective wisdom of the multinational crowd for developing high-quality software with a minimal consumption of time and cost [13–15]. Figure 1 represents various virtual crowd participations in global software development.

The global software development has emerged as an important method to ensure optimum resources [15]. The development is carried out in a distributed manner and on numerous sites that may be located on various demographical settings. A multinational crowd team participates in the development of software [16]. The distribution of tasks may depend on the logical fragmentation of various project components and modules in global software development. The fragmentation of components or modules of software will enable organizations to delegate task to global crowd teams. The skills of the global crowd would increase the production and quality of software. By utilizing the GSD approach to software development, multiple modules or projects will be completed in parallel [17].

Globalization has its perks, but it still has pitfalls. As an advantage, you achieve time-zone efficiency in different regions that decrease the associated costs. Working on a globally distributed task has created a wide variety of problems for software developers, driven by market and resource requirements. The development may be associated with planning and coordinating people as well as addressing language and cultural barriers. It also breeds distrust, as costlier engineers (fearful of losing their jobs) are forced to train their much less expensive counterparts. Communication and teamwork problems have always been daunting in large software engineering ventures [10, 12].

Large projects may entail a high degree of knowledge integration and the synchronized determinations of several developers [18]; therefore, an efficient selection of the virtual crowd is required for enhancing the quality of software. The selection process in global software development shifts is a complex task. The manual selection that is based only on expert judgement may lead to inappropriate selection of team members. To make adequate selection, there is a significant need for a criteria-based crowd selection [9]. The aim of this research is to select an effective virtual crowd for the development of quality software. The crowd will be selected according to their unique features such as their skills, experiences, expertise, and knowledge.

2. Literature Review

Software development consists of people who perform various activities by utilizing a range of tools, approaches, and techniques [19]. The activities may include deep analysis, designing the overall processes, writing codes, and testing the end product [20]. The activities are linked for development and operating of software. For accomplishment of these activities, a number of individuals are called [21]. Startup companies face a major challenge in recruiting professionals with the right skills for development of software [22].

2.1. Selection Based on Skills. An individual strives to learn appropriate and specific skills [23]. Firms are outsourcing software developmental tasks to lower wage countries. Highly skilled labors with technical skills are recruited to accomplish tasks [24]. Software are outsourced to highly skilled workforce [25]. Software engineering usually separates skill into “hard skill” and “soft skill.” Hard skills are the technical and knowledge specifications an individual needs to perform a task. It means that a person should successfully carry out the planned task on theoretical foundations and realistic exposure. Soft skills are “the cluster of personality traits, social graces, language skills, friendliness, and optimism that mark each one of us to varying degrees.” Soft skills are based on psychology and cover a wide spectrum of features including styles of personalities, abilities for social interactions, communication, and personal traits. In the workplace, people tend to weigh soft skills which integrate hard skills. Soft skills are the individuality that has a significant influence on a person’s actions when engaging with others in a workplace. The possession and use of soft skills increase the probability of success and help to achieve the project’s common goal in the collaborative world of software development [21].

2.2. Social and Behavioral-Based Selection. The success or failure of a project can be determined by individual team members. The social and behavioral attributes of the team members also impact the performance of group-enabled activity [26]. A participant may behave socially to attain affiliation (such as social comparison, emotional provisions, progressive motivation, and devotion) or to achieve (personal) satisfaction on completing tasks [27, 28].

2.3. Selection Based on Past Performance. Each individual has a personal history that dictates his own perceptions and behavior in social environments [29].

2.4. Selection Based on Geographical Locations. Globally distributed software development is carried out by collaborative teams for accomplishing the goals of the project.

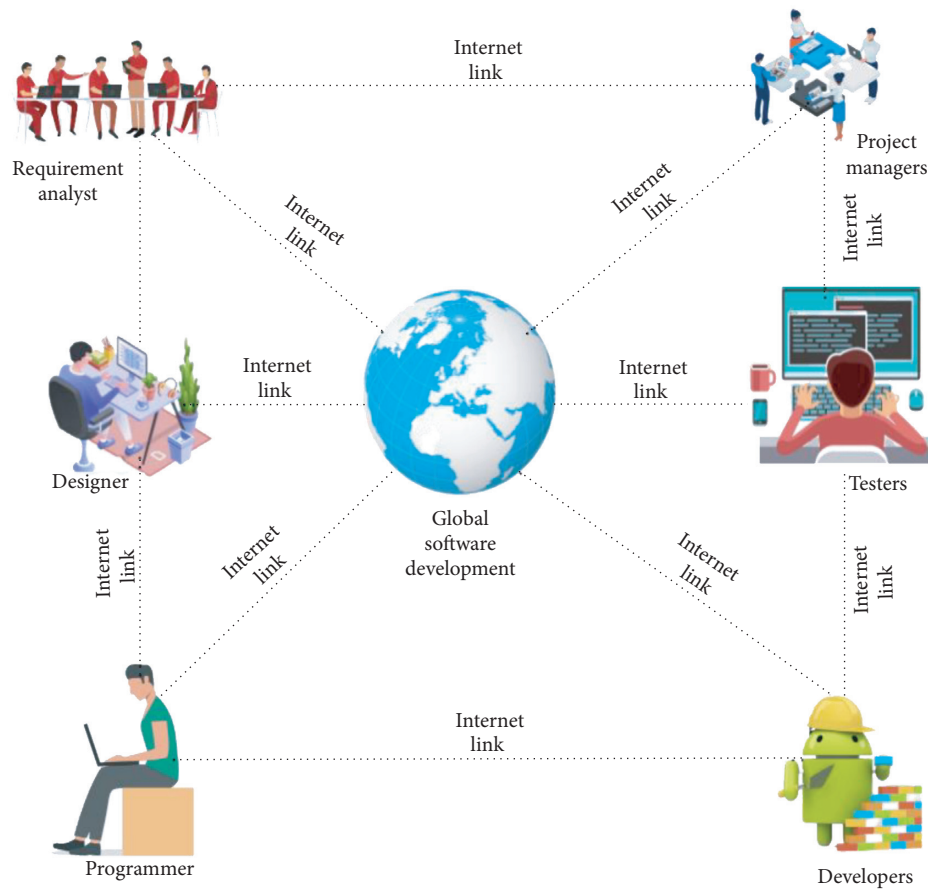


FIGURE 1: Global software developmental phenomenon.

These teams are diverse and dispersed on different geographical locations [30]. Dispersed and diverse crowd teams consist of large labor pool having various skills which ultimately decrease the developmental time and cost [31].

2.5. Recommendation-Based Team Allocation for Software Development. Software development is a complex process which requires the appropriate method and financial and human resources (HR). Software engineering, therefore, focuses on how the complexity and effort required for software development activities can be reduced. A recommendation framework for allocating global software teams was proposed. This framework helps the management teams of global software development that are participating in the software implementation phase; it recommends the appropriate crowd teams to complete various components of software [32].

2.6. Trust-Based Selection. Trust is an important element to determine virtual teams' success and failure for global software development [33]. "Virtuality requires trust to make it work: technology on its own is not enough" [34]. It is, therefore, logical to say that trust is vital for GSD teams as well. Trust in virtual teams has long been studied, and the effect has been well known. Virtual teams with a high level of trust experience important social interaction, reliable

patterns of communication, strong reviews, good leadership, excitement, and willingness to face technological uncertainty. Trust is a key to efficient mutual improvement and is therefore essential for successful teamwork, which is essential for cooperation and productivity [35].

3. Methodology

The Internet has transformed the world into a global village. People use the Internet to engage in and contribute to a variety of events. Global software development takes place all over the world through the Internet, and a large number of people participate in the process. In global software development environments, selecting suitable virtual crowds is a challenging task. The crowd will be chosen based on their distinct characteristics. These characteristics were noticed through reviewing the literature. As a large number of heterogeneous and complex people are involved in global software development, the crowd's characteristics may be diverse, and there may be redundancy that will affect proper selection. These were filtered out to remove duplication and ambiguity.

3.1. Characteristic-Based Filtering. Virtual crowd on Internet consists of heterogeneous people that exhibit various characteristics; therefore, a characteristic-based virtual

crowd selection is essential to solve global world. The selection of the appropriate crowd participant in global software development is addressed in our method by considering their unique characteristics. Various characteristics that have been identified in literature studies are given in Table 1 for the purpose to select or reject the crowd based on these characteristics. A complex dataset of these characteristics was obtained in our evaluation process that was reduced by eliminating less appropriate characteristics. The elimination of less appropriate characteristics will increase scalability and efficiency. A subset from the original set of virtual crowd V is obtained using the characteristics-based virtual crowd selection method. However, the precision of original virtual crowd set (V) was not altered. Our crowd selection method would select the crowd based on their best characteristics.

3.2. Ant Colony Optimization. Ant colony optimization was first developed as a swarm intelligence strategy by Dorigo and Blum in 1990 [42]. Swarm intelligence is a problem-solving approach focused on the unique interaction of animals and insects that are relatively new. Ants have inspired a wide range of methods and techniques; the most well-known and common of which is ant colony optimization, a general-purpose optimization technique. Ant colony optimization (ACO) is a concept that is focused on the scavenging behavior of various ant species. When ants move from one location to another, they drop pheromones (chemicals) on the surface to mark welcoming directions for other ants in the colony (members). Ant colony optimization uses a similar mechanism to solve optimization issues. To find an optimal solution, the ant colony optimization method is primarily used [43]. Ants solve problems using two factors: heuristic knowledge and frequency of pheromones. Artificial ants may communicate with each other to produce high-quality results. The values of pheromone trails are retrieved by indirect contact (sensed the pheromone) between different ants. Ants do not change; rather, they adjust how other ants represent and interpret the problem [43]. For implementation of the characteristic-based virtual crowd selection (CBVCS) method, ACOs' technique is applied which will solve dilemma of virtual crowd selection in the area of global software development.

3.3. Characteristic-Based Virtual Crowd Selection (CBVCS) Method. In global software growth, the ACO model would be used to deal with the problem of virtual crowd selection. The ACO algorithm for virtual crowd selection optimization is implemented in several steps. The proposed method will address the virtual crowd selection problem. The selection of a virtual crowd starts with the formation of ants that will travel along different paths (edges) and pick a crowd based on the pheromone value present on each edge. If the ant traversal reaches a stopping criterion, the ants will stop (the traversal will end), and the best subset of the virtual crowd will be formed, and these virtual crowds will be utilized for software development tasks. The pheromone materials are transformed, and the loop is restarted if the traversal fails to

TABLE 1: Various characteristics identified in literature studies are given for the purpose to select or reject the crowd based on these characteristics.

| S. no. | Features of virtual crowd | Citation |
|--------|---------------------------|-----------------|
| 1 | Skills | [21, 23–25, 33] |
| 2 | Knowledge | [26] |
| 3 | Collaborative | [35–37] |
| 4 | Trust | [33–35] |
| 5 | Cooperative | [33] |
| 6 | Coordinative | [37, 38] |
| 7 | Visibility | [11] |
| 8 | Experience | [23, 39] |
| 9 | Educated | [40] |
| 10 | Decision maker | [41] |

follow the stopping criteria. The flow of characteristic-based virtual crowd selection is represented in Figure 2.

Virtual crowds on the Internet show a variety of distinct characteristics that have been identified in existing literature (Table 1). By integrating the appropriate and less appropriate characteristics, a virtual crowd set (V) is generated (Table 2). The original set will be reduced by removing the inappropriate characteristic using the characteristics-based virtual crowd selection method, while the higher degree of precision in the depiction of the set will be retained. As a result, only a portion of the virtual crowds is selected. The choice of potential virtual crowd is unaffected by the previous virtual crowd attached to a node. It is not necessary, however, that the subset created be of equal size. The virtual crowd selection problem is mapped using the steps below.

- (a) Representation in the graphical structure
- (b) Heuristic and pheromone-based selection
- (c) Pheromone modification
- (d) Evaluating results

3.3.1. Representation in Graphical Structure. In ant colony optimization technique, the problem is represented in the graphical structure (Figure 3). The nodes represent various crowds, while the edges indicate the corresponding virtual crowd decision. The nodes are connected together, allowing any virtual crowd to be selected. An optimal subset of the virtual crowd is chosen when an ant traverses the graph or visits various nodes. The ant traversal must meet the stopping requirements (select optimal virtual crowd). In Figure 3, the ants $K, L, M, N, O, P, Q, R, S,$ and T are free to leave their nest and travel to different nodes such as $V1, V2, V3, V4, V5, V6, V7, V8, V9,$ and $V10$. On the edge-to-edge traversal process, these ants drop pheromones, a chemical substance (see Figure 4). Other ants follow the pheromones and switch in response to the probability of pheromone levels on different edges, i.e., if pheromone levels are high, the ant will choose only high pheromone value edges (bold line) and only those specific nodes. Using the transition rules, the ant K from the nest will choose node $V1$ and then $V4, V5, V8,$ and $V9$ are then selected. As the ant traversal hits $V9$, it meets the stopping criteria and halts, presenting a

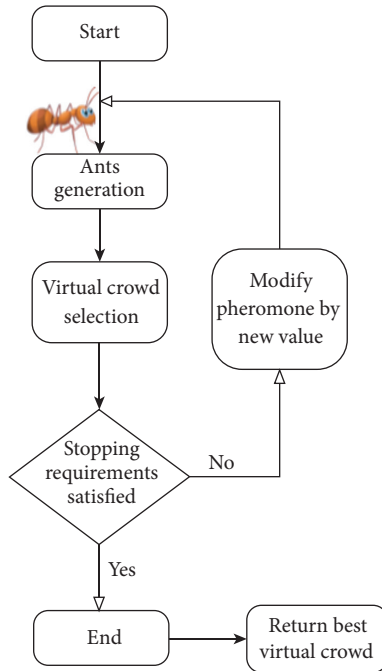


FIGURE 2: The proposed characteristic-based virtual crowd selection (CBVCS) method.

TABLE 2: Virtual crowds’ set.

| Virtual crowds | Characteristics of virtual crowds |
|----------------|-----------------------------------|
| V1 | Skills |
| V2 | Cooperative |
| V3 | Visible |
| V4 | Experience |
| V5 | Trust |
| V6 | Coordinative |
| V7 | Knowledge |
| V8 | Decision maker |
| V9 | Collaborative |
| V10 | Education |

partial solution to the original virtual crowd set “V,” which includes virtual crowd V1, V4, V5, V8, and V9 (Figure 5). As a consequence, there is a high degree of accuracy. Following that, the virtual crowd subset is used as a candidate for software development tasks.

3.3.2. Heuristic and Pheromone-Based Selection. The best characteristics are being used to evaluate virtual crowds. A simple meta-analysis local search mechanism is used to find virtual crowd substitutes at first. The (η_i) heuristic variable is used in combination mostly with the pheromone function in the ACO algorithm to make a proper transition. The best virtual crowds’ subset is found by determining the pheromone and heuristic value. The appropriate virtual crowd is assigned due to the higher pheromone value. Inappropriate virtual crowds, on the contrary, are rejected due to the lower pheromone value associated with the relevant edge. An ant present on V1 node chooses whether V3 would be chosen or not, and the choice is associated with the possibility of the

maximum pheromones on the paths (edges). This possibility of pheromone is determined by means of the following formula:

$$P(\text{edge}) = \frac{P(\text{pheromones}(Xi)\eta_i)}{\sum(P(\text{pheromones}(Xi))\eta_i)} \quad (1)$$

The chance of selecting a node is calculated using equation (1), where probability is expressed by P , edges or paths are expressed by Xi , and heuristic strength is indicated by η_i . The η_i value should be kept greater if a path is to be chosen; otherwise, it must be kept lesser. The traversal and selection of a node are influenced by the pheromone value. The ant will move along the pheromone-rich edge.

3.3.3. Pheromone Modification. If the ant traversal condition is not met, the pheromone values are transformed, new ants are created, and the cycle repeats. Using the equation, the pheromone of ants is modified (B):

$$\tau_1(\tau + 1) = (1 - \rho) \cdot \tau_1(\tau) + \rho \cdot \Delta\tau_1(\tau). \quad (2)$$

In the above equation, ρ is the decay coefficient of pheromone, whose value is in a range of 0-1, τ_1 is the remaining pheromone amount that are remaining on edge linking nodes, and $\Delta\tau_1$ is the pheromone increment for subsequent progression. Best ants deposit many pheromones on optimal solution nodes, revealing optimal virtual crowd characteristics as a result.

3.3.4. Evaluating Results. The process for the CBVCS starts with the development of an arbitrary number of artificial ants. The ants are arranged on the graph with their numbers equaling the number of virtual crowds that are associated. Each ant begins the process of building a graph from a single (virtual crowd) node by traversing it. Ant travels from a starting point in a probabilistic manner, crossing various nodes before reaching the stopping condition. The virtual crowd set is then gathered and analyzed for an optimal subset. When the best virtual crowds have been found, the experiment is almost over, and the outcomes are exposed (Figure 5). When the halting requirements are not met, a pheromone modification takes place, a new ant colony is formed, and the process starts all over again.

Appropriate characteristics will be sorted out from Table 2 using characteristics filtering.

4. Results and Discussion

Digital teams are rapidly driving the need for fast-to-market, low-cost, and rapid solutions to complex organizational challenges as the economy progresses. Organizations may use virtual teams to integrate the expertise and talents of workers and nonemployees, while reducing time and space barriers. To increase their efficiency and productivity, companies are now heavily investing in virtual teams. Global software development has emerged as a key strategy for ensuring optimal resource utilization. The development is carried out in a dispersed manner across a number of

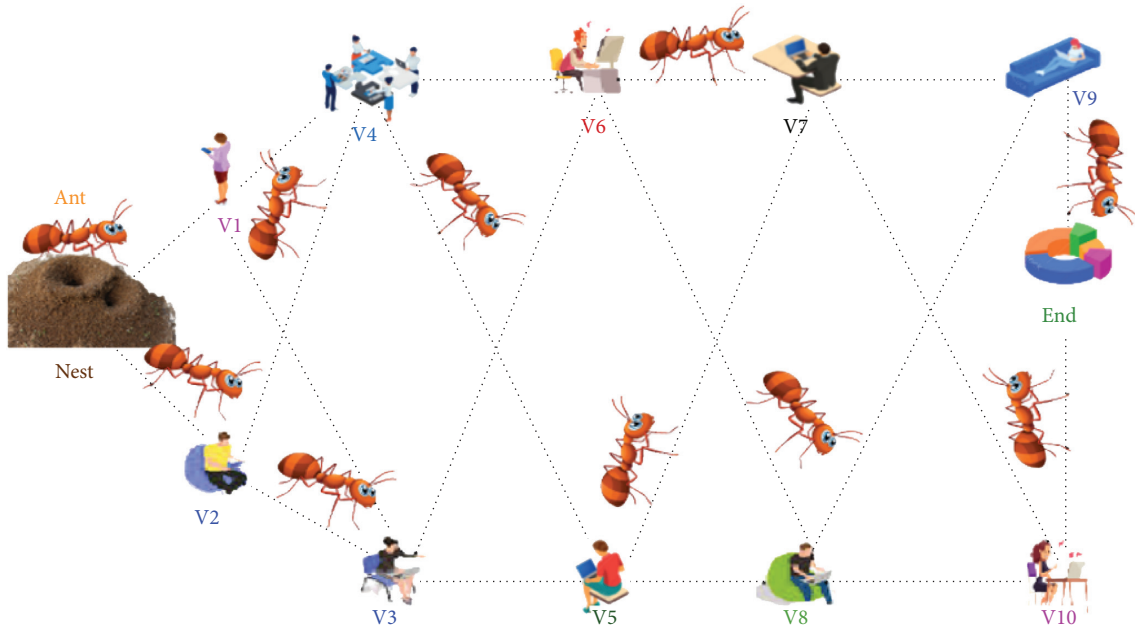


FIGURE 3: Path traversal of various ants.

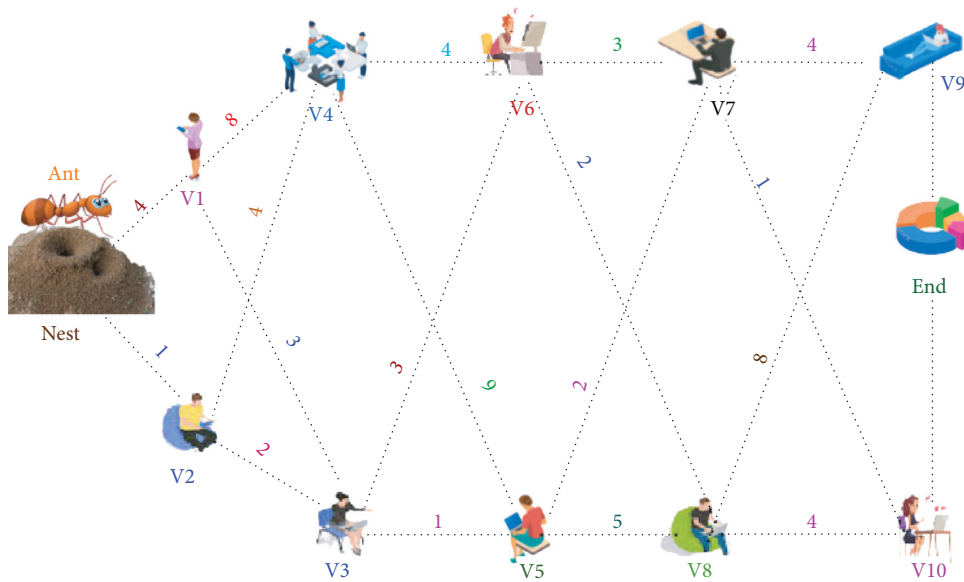


FIGURE 4: Ants leaving pheromones on traversed paths.

locations, each of which may be situated in a different demographical environment. In the development of the global software, a multinational crowd team is involved. When selecting virtual crowds, care must be taken, as the standard of software increases with the selection of appropriate virtual crowds. In our system for virtual crowd selection based on characteristics, a virtual crowd set (10 virtual crowds “V1, V2, V3, V4, V5, V6, V7, V8, V9, and V10”) with multiple characteristics (relevant and irrelevant) is placed in a graphical structure. Virtual crowds are depicted as a group of nodes connected by edges; an equal number of ants are generated to navigate along multiple edges and pick virtual

crowds, resulting in a partial solution to the virtual crowd set. The ants will avoid traversing and produce the best virtual crowds’ subset with multiple characteristics if a partial solution meets the stopping criterion (i.e., it selects the best virtual crowds’ subset *D*). If the ants fail to follow the stopping criteria, the pheromones are changed, and the process starts over. The selection and rejection of the virtual crowd is determined by the likelihood of pheromone across each edge; if the value is higher, the nodes will be selected and the best virtual crowds’ subset will be created in ant’s traversal; if the value is lower, the edges and the connected virtual crowd node will be rejected. Table 3 shows the ants

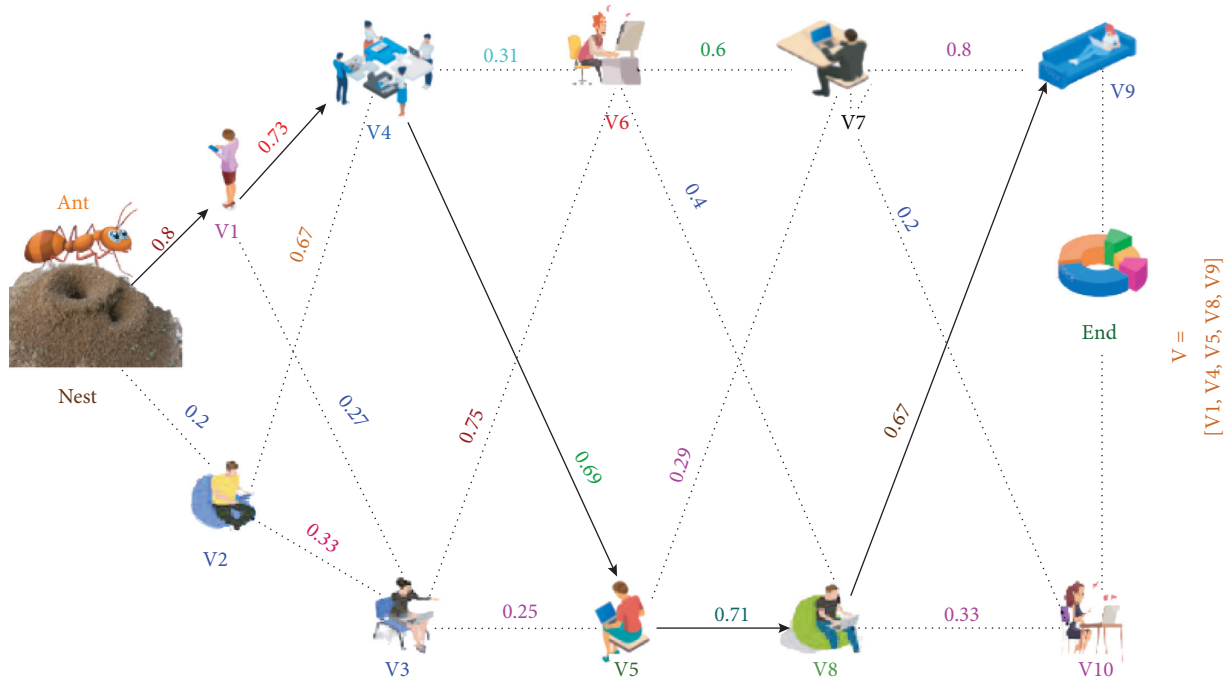


FIGURE 5: Pheromone probabilitywise virtual crowd selection.

TABLE 3: Ants and their preferred direction.

| Virtual crowd | Selected virtual crowd | Ants | Pheromone (value) on selected edges | Paths selection probability node/ \sum (value) connected edges | Path selection probability of an ant (from starting node to last node) $\sum P$ (pheromone on edge) | Probability pheromone P (pheromone on selected edges)/ $\sum P$ (pheromones on all edges) |
|---------------|-------------------------|----------|-------------------------------------|--|---|---|
| V1 | V1, V4, V5, V8, and V9 | <i>K</i> | 4, 8, 9, 5, and 8 | 0.8, 0.73, 0.69, 0.71, and 0.67 | 3.6 | 0.14 |
| V2 | V2, V3, V5, V8, and V9 | <i>L</i> | 1, 2, 1, 5, and 8 | 0.2, 0.33, 0.25, 0.71, and 0.67 | 2.16 | 0.08 |
| V3 | V1, V3, V6, V7, and V10 | <i>M</i> | 4, 3, 3, 3, and 1 | 0.8, 0.27, 0.75, 0.6, and 0.2 | 2.62 | 0.10 |
| V4 | V2, V3, V6, V8, and V9 | <i>N</i> | 1, 2, 3, 2, and 8 | 0.2, 0.33, 0.75, 0.4, and 0.67 | 2.35 | 0.09 |
| V5 | V2, V3, V6, V7, and V9 | <i>O</i> | 1, 2, 3, 3, and 4 | 0.2, 0.33, 0.75, 0.6, and 0.8 | 2.68 | 0.10 |
| V6 | V1, V4, V6, V8, and V10 | <i>P</i> | 4, 8, 4, 2, and 4 | 0.8, 0.73, 0.31, 0.4, and 0.33 | 2.57 | 0.10 |
| V7 | V2, V4, V6, V8, and V10 | <i>Q</i> | 1, 4, 4, 2, and 4 | 0.2, 0.67, 0.31, 0.4, and 0.33 | 1.91 | 0.07 |
| V8 | V1, V3, V5, V7, and V9 | <i>R</i> | 4, 3, 1, 2, and 4 | 0.8, 0.27, 0.25, 0.29, and 0.8 | 2.41 | 0.09 |
| V9 | V1, V4, V5, V7, and V9 | <i>S</i> | 4, 8, 9, 2, and 4 | 0.8, 0.73, 0.69, 0.29, and 0.8 | 3.31 | 0.13 |
| V10 | V2, V3, V6, V8, and V10 | <i>T</i> | 1, 2, 3, 2, and 4 | 0.2, 0.33, 0.75, 0.4, and 0.33 | 2.01 | 0.08 |
| | | | | | $\sum = 25.62$ | |

and their preferred direction. To find the best traversing path, the probability of edges is determined, which then selects only suitable virtual crowds. In our proposed virtual crowd selection process, the virtual crowds' subset V1, V4,

V5, V8, and V9 were selected by traversal of ant *K* because the probability of the values of pheromone on their respective edges linking the nodes (virtual crowds) is greater than other edges concerning other nodes (virtual crowds).

5. Conclusion

Innovations in technologies have significantly reshaped the software development environment. With the wide applications of Web 3.0 and improvements in cloud computation, the software of this era are built with the help of virtual crowd present on various sites on Internet. Globalization is considered as a major trend especially among software developing industries. National markets are constantly transformed into global markets by economic forces, generating new forms of competition and cooperation across national borders. For technological firms, global software development (GSD) is increasingly becoming the trend. The organization utilizes the collective wisdom of the multinational crowd for developing high quality software with a minimal consumption of time and cost. The development is carried out in a distributed manner and on numerous sites that may be located on various demographical settings. A multinational crowd team participates in the development of a software. The presented research focuses on the virtual crowds' selection problem in global software development. Characteristics-based virtual crowd selection using ant colony optimization for the virtual crowd selection problem in global software development was proposed which selects appropriate virtual crowds to carry out software developmental tasks. The key contribution of our research is to select the appropriate virtual crowd based on the multicriteria characteristics. The selection of the appropriate virtual crowd will increase the efficiency and effectiveness of global software development. In this article, the method is theoretically presented with fewer virtual crowds. We would use it in the future because it is effective (i.e., it selects virtual crowds based on multicriteria characteristics), and it will play an important role in the virtual crowd selection process in global software development.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] V. Casey, "Imparting the importance of culture to global software development," *ACM inroads*, vol. 1, no. 3, pp. 51–57, 2011.
- [2] R. Sangwan, M. Bass, N. Mullick, D. J. Paulish, and J. Kazmeier, *Global Software Development Handbook*, CRC Press, Boca Raton, FL, USA, 2006.
- [3] A. Kankanhalli, B. C. Y. Tan, and K.-K. Wei, "Conflict and performance in global virtual teams," *Journal of Management Information Systems*, vol. 23, no. 3, pp. 237–274, 2006.
- [4] N. Ale Ebrahim, S. Ahmed, and Z. Taha, "Virtual teams: a literature review," *Australian journal of basic and applied sciences*, vol. 3, no. 3, pp. 2653–2669, 2009.
- [5] M. Faisal Abrar, M. Sohail, A. Sikandar et al., "De-motivators for the adoption of agile methodologies for large-scale software development teams," *An SLR from management perspective*, vol. 32, no. 12, Article ID e2268, 2020.
- [6] R. A. Khan, M. Y. Idris, S. U. Khan et al., "An evaluation framework for communication and coordination processes in offshore software development outsourcing relationship: using fuzzy methods," *IEEE Access*, vol. 7, pp. 112879–112906, 2019.
- [7] M. Yaseen, S. Baseer, S. Ali, and S. U. Khan, "Requirement implementation model (RIM) in the context of global software development," in *Proceedings of the International Conference on Information and Communication Technologies (ICICT)*, Karachi, Pakistan, December 2015.
- [8] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Software*, vol. 18, no. 2, pp. 16–20, 2001.
- [9] M. Goyal and C. Gupta, "Intuitionistic fuzzy decision making towards efficient team selection in global software development," *Journal of Information Technology Research*, vol. 13, no. 2, pp. 75–93, 2020.
- [10] C. Ebert and P. De Neve, "Surviving global software development," *IEEE Software*, vol. 18, no. 2, pp. 62–69, 2001.
- [11] J. A. Espinosa and E. Carmel, "The impact of time separation on coordination in global software teams: a conceptual foundation," *Software Process: Improvement and Practice*, vol. 8, no. 4, pp. 249–266, 2003.
- [12] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "An empirical study of global software development: distance and speed," in *Proceedings of the 23rd International Conference on Software Engineering. ICSE*, May 2001.
- [13] N. Ali and R. Lai, "Global software development: a review OF its practices," *Malaysian Journal of Computer Science*, vol. 34, no. 1, pp. 82–129, 2021.
- [14] A. A. Khan and M. A. Akbar, "Systematic literature review and empirical investigation of motivators for requirements change management process in global software development," *Journal of Software: Evolution and Process*, vol. 32, no. 4, Article ID e2242, 2020.
- [15] J. A. Khan, S. U. R. Khan, J. Iqbal, and I. U. Rehman, "Empirical investigation about the factors affecting the cost estimation in global software development context," *IEEE Access*, vol. 9, pp. 22274–22294, 2021.
- [16] M. A. Akbar, A. Gumaei, A. Al-Sanad, and M. Shafiq, "Towards efficient and secure global software development using blockchain," in *Proceedings of the Evaluation and Assessment in Software Engineering*, pp. 493–498, Copenhagen, Denmark, April 2020.
- [17] M. W. Bhatti and A. Ahsan, "Effective communication among globally distributed software development teams: development of an "effective communication" scale," in *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming*, pp. 2014–2039, IGI Global, Pennsylvania, PA, USA, 2021.
- [18] K. Crowston, Q. Li, K. Wei, U. Y. Eseryel, and J. Howison, "Self-organization of teams for free/libre open source software development," *Information and Software Technology*, vol. 49, no. 6, pp. 564–575, 2007.
- [19] G. Maturro, "Soft skills in software engineering: a study of its demand by software companies in Uruguay," in *Proceedings of the 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, IEEE, San Francisco, CA, USA, May 2013.
- [20] F. Ahmed, L. F. Capretz, S. Bouktif, and P. Campbell, "Soft skills requirements in software development jobs: a cross-cultural empirical study," *Journal of Systems and Information Technology*, vol. 14, 2012.

- [21] F. Ahmed, L. F. Capretz, S. Bouktif, and P. Campbell, "Soft skills and software development: a reflection from the software industry," 2015, <https://arxiv.org/abs/1507.06873>.
- [22] P. Mangiza and I. Brown, "Requisite skills profile of software development professionals for startups," in *Proceedings of the Conference of the South African Institute of Computer Scientists and Information Technologists*, Cape Town, South Africa, September 2020.
- [23] S. L. Adams and V. Anantatmula, "Social and behavioral influences on team process," *Project Management Journal*, vol. 41, no. 4, pp. 89–98, 2010.
- [24] S. Matthiesen and P. Bjørn, "When distribution of tasks and skills are fundamentally problematic," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, pp. 1–16, 2017.
- [25] P. Bjørn, A.-M. Søderberg, and S. Krishna, "Translocality in global software development: the dark side of global agile," *Human-Computer Interaction*, vol. 34, no. 2, pp. 174–203, 2019.
- [26] S. A. Licorish and S. G. MacDonell, "Communication and personality profiles of global software developers," *Information and Software Technology*, vol. 64, pp. 113–131, 2015.
- [27] C. A. Hill, "Affiliation motivation: people who need people. . . but in different ways," *Journal of Personality and Social Psychology*, vol. 52, Article ID 1008, 1987.
- [28] J. E. Mathieu, "A test of subordinates' achievement and affiliation needs as moderators of leader path-goal relationships," *Basic and Applied Social Psychology*, vol. 11, no. 2, pp. 179–189, 1990.
- [29] B. Simon and T. F. Pettigrew, "Social identity and perceived group homogeneity: evidence for the ingroup homogeneity effect," *European Journal of Social Psychology*, vol. 20, no. 4, pp. 269–286, 1990.
- [30] H. Holmstrom, E. O. Conchuir, P. J. Agerfalk, and B. Fitzgerald, "Global software development challenges: a case study on temporal, geographical and socio-cultural distance," in *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE'06)*, October 2006.
- [31] E. Ó Conchúir, H. Holmström Olsson, P. J. Ågerfalk, and B. Fitzgerald, "Benefits of global software development: exploring the unexplored," *Software Process: Improvement and Practice*, vol. 14, no. 4, pp. 201–212, 2009.
- [32] T. A. B. Pereira, V. S. dos Santos, B. L. Ribeiro, and G. Elias, "A recommendation framework for allocating global software teams in software product line projects," in *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*, Cape Town, South Africa, May 2010.
- [33] A. Iftikhar, M. Alam, S. Musa, and M. M. Su'ud, "Trust development in virtual teams to implement global software development (GSD): a structured approach to overcome communication barriers," in *Proceedings of the IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS)*, August 2017.
- [34] V. Casey, "Developing trust in virtual software development teams," *Journal of theoretical and applied electronic commerce research*, vol. 5, no. 2, pp. 41–58, 2010.
- [35] N. B. Moe and D. Šmite, "Understanding a lack of trust in global software teams: a multiple-case study," *Software Process: Improvement and Practice*, vol. 13, no. 3, pp. 217–231, 2008.
- [36] F. Lanubile, C. Ebert, R. Prikladnicki, and A. Vizcaino, "Collaboration tools for global software engineering," *IEEE software*, vol. 27, no. 2, pp. 52–55, 2010.
- [37] I. Steinmacher, A. P. Chaves, and M. A. Gerosa, "Awareness support in global software development: a systematic review based on the 3C collaboration model," in *Proceedings of the International Conference on Collaboration and Technology*, September 2010.
- [38] J. A. Espinosa, S. A. Slaughter, R. E. Kraut, and J. D. Herbsleb, "Team knowledge and coordination in geographically distributed software development," *Journal of Management Information Systems*, vol. 24, no. 1, pp. 135–169, 2007.
- [39] A. Mockus and D. M. Weiss, "Globalization by chunking: a quantitative approach," *IEEE software*, vol. 18, no. 2, pp. 30–37, 2001.
- [40] J. M. Verner, O. P. Brereton, B. A. Kitchenham, M. Turner, and M. Niazi, "Risks and risk mitigation in global software development: a tertiary study," *Information and Software Technology*, vol. 56, no. 1, pp. 54–78, 2014.
- [41] A. Avritzer, D. Paulish, Y. Cai, and K. Sethi, "Coordination implications of software architecture in a global software development project," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1881–1895, 2010.
- [42] M. Dorigo and C. Blum, "Ant colony optimization theory: a survey," *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 243–278, 2005.
- [43] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.